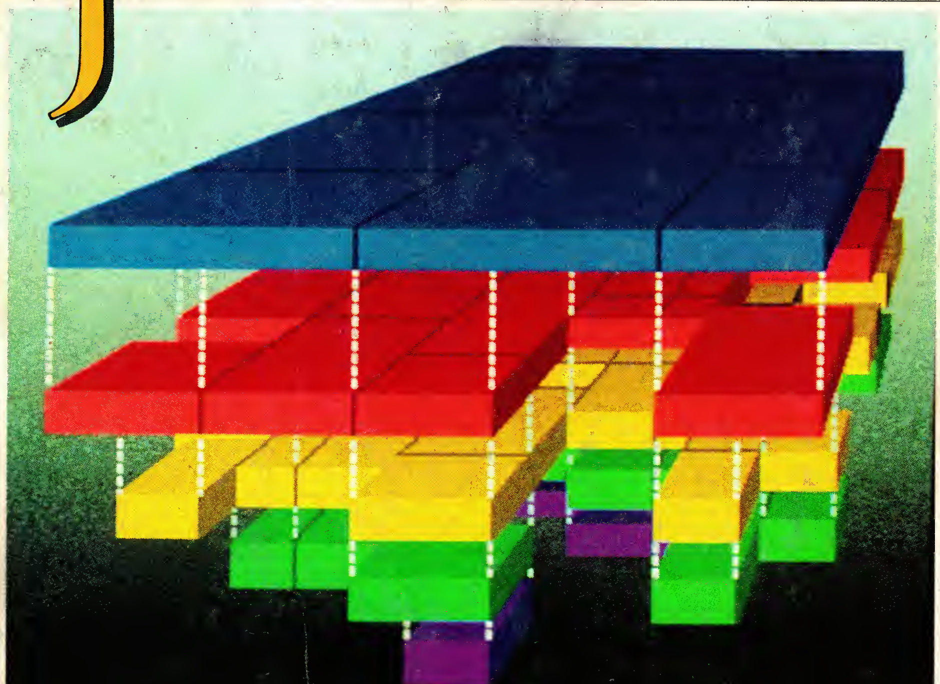
SEPTEMBER 1986

VOL. 4, NO. 9 $3.95

FOR IBM PERSONAL COMPUTER USERS

PC TECH JOURNAL®



FORTH
Architecture and Innermost Workings

INTERACTIVE CAD

CLARION REVIEW

DATA MANAGER: DATAEASE

Turbo GameWorks™

Also recently released, Turbo GameWorks is what you think it is: "Games" and "Works." Games you can play right away (like Chess, Bridge and Go-Moku), plus the Works—which is how computer games work. All the secrets and strategies of game theory are there for you to learn. You can play the games "as is" or modify them any which way you want. Source code is included to let you do that, and whether you want to write your own games or simply play the off-the-shelf games, Turbo GameWorks will give hours of diversion, education, and intrigue. George Koltanowski, Dean of American Chess, and former President, United States Chess Federation, reacted to Turbo GameWorks like this, "*With Turbo GameWorks, you're on your way to becoming a master chess player,*" and Kit Woolsey, writer, author, and twice Champion of the Blue Ribbon Pairs, wrote, "*Now play the world's most popular card game—Bridge . . . even program your own bidding or scoring conventions.*" Suggested retail: $69.95.

Turbo Database Toolbox™

A perfect complement to Turbo Pascal, because it contains a complete library of Pascal procedures that allows you to search and sort data and build powerful database applications. Having Turbo Database Toolbox means you don't have to reinvent the wheel each time you write a Turbo Pascal program. It comes with source code for a free sample database—right on disk. The database can be searched by keywords or numbers. Update, add, or delete records as needed. Just compile it and it's ready to go to work for you. Suggested retail: $69.95.

Turbo Graphix Toolbox™

It includes a library of graphics routines for Turbo Pascal programs. Lets even beginning programmers create high-resolution graphics with an IBM, Hercules,™ or compatible graphics adapter. Our Turbo Graphix Toolbox includes all the tools you'll ever need for complex business graphics, easy windowing, and storing screen images to memory. It comes complete with source code, ready to compile. Suggested retail: $69.95.

Technical Specifications:

TURBO PASCAL 3.0 Minimum memory:128K; includes 8087 and BCD features for 16-bit MS-DOS and CP/M-86 systems. CP/M-80 version minimum memory: 48K; 8087 and BCD features not available

TURBO DATABASE TOOLBOX Minimum memory: 128K. CP/M-80 minimum memory: 48K. Requires Turbo Pascal 2.0 or later.

TURBO GRAPHIX TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later, Turbo Pascal 3.0, and IBM CGA, Hercules Monochrome Card or equivalent.

TURBO TUTOR 2.0 Minimum memory: 192K. CP/M-80 version minimum memory 48K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO EDITOR TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO GAMEWORKS* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO PROLOG* Minimum memory: 384K.

REFLEX: THE ANALYST* Minimum memory: 384K. Requires IBM CGA, Hercules Monochrome Card or equivalent. Works with Intel's AboveBoard-PC and -AT; AST's RAMpage! and RAMpage! AT; Quadram's Liberty-PC and -AT; Tecmar's 640 Plus; IBM's EGA and 3270/PC; AT&T's 6300 and many others.

REFLEX WORKSHOP* Minimum memory: 384K. Requires Reflex: The Analyst. Two disk drives or hard disk recommended.

TURBO LIGHTNING* Minimum memory: 256K. Two disk drives required. Hard disk recommended.

LIGHTNING WORD WIZARD* Minimum memory: 256K. Requires Turbo Lightning. Turbo Pascal 3.0 required to edit source code.

SIDEKICK* Minimum memory: 128K.

TRAVELING SIDEKICK* Minimum memory: 256K.

SUPERKEY* Minimum memory: 128K.

*For IBM PC, AT, XT, PCjr and true compatibles only, running PC/MS-DOS 2.0 or later.

Recognition for Borland International has come from business, trade, and media, and includes both product awards and awards for technical excellence and marketing. Borland was named "Company of the Year" by PC Magazine; Sidekick, the #1 best seller for the IBM PC, was named "Product of the Year" by InfoWorld; Turbo Pascal was selected one of PC Week's Top 10 Products for 1984; SuperKey won one of PC Magazine's "Best of 1985" awards; Reflex, The Analyst was recognized in the "Software Products of the Year" awards by InfoWorld; and Reflex and SideKick were both nominated for British Micro Awards in 1986.

COMPLETELY NEW VERSION!

Turbo Tutor® 2.0

Just released (July '86), the new Turbo Tutor can take you from "What's a computer?" on through to complex data structures, assembly languages, trees, tips on writing long programs in Turbo Pascal, and a high level of expertise. Source code for everything is included. New split screens allow you to put source text in the bottom half of the screen and run the examples in the top half. There are quizzes that ask you, show you, tell you, teach you. You get a 450-page manual—which is not as daunting as it sounds, because unlike many software manuals, it was not written by orangutans. (With our all "almost-free" upgrade, you can upgrade to Turbo Tutor 2.0 by sending us your master diskettes, proof of purchase, and $10.00, which covers shipping and handling.) Suggested retail: $39.95.

NEW SPECIAL!

Turbo Pascal® 3.0

"*For the IBM PC, the benchmark Pascal compiler is undoubtedly Borland International's Turbo Pascal,*" says Gary Ray of *PC Week.* We and more than 500,000 other people around the world think Mr. Ray got that right. Since launch, Turbo Pascal has become the *de facto* worldwide standard in high-speed Pascal compilers. Described by Jeff Duntemann of *PC Magazine* as the "*Language deal of the century,*" Turbo Pascal is now an even better deal than that—because we've *included* the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95! You now get a lot more for a lot less: the compiler, a completely integrated programming environment, and BCD reals and 8087 support—all for a suggested retail of only $99.95.

Turbo Editor Toolbox™

Recently released, we called our new Turbo Editor Toolbox a "construction set to write your own word processor." Peter Feldmann of *PC Magazine* covered it pretty well with, "*A 'write your own word processor' program for intermediate level programmers, with lots of help in the form of prewritten procedures covering everything from word wrap to pull-down windows.*" Source code is included, and we also include MicroStar, a full-blown text editor with pull-down menus and windowing. It interfaces directly with Turbo Lightning to let you spell-check your MicroStar files. Jerry Pournelle of *BYTE* magazine said, "*The new Turbo Editor Toolbox is the Turbo Pascal source code to just about anything you ever wanted a PC-compatible text editor to do.*" Suggested retail: $69.95.

Borland's Business Productivity Programs:

Reflex: The Analyst™ Analytical database manager. Provides complete new look at data normally hidden by programs like 1-2-3® and dBASE.® Best report generator for 1-2-3.

NEW! **Reflex Workshop™** Important new addition to Reflex: The Analyst. Gives you 22 different templates to run your business right.

SideKick® Complete RAM-resident desktop management includes notepad, dialer, calculator and more.

Traveling SideKick™ Electronic version of business/personal diaries, daytime organizers; works with your SideKick files; important professional tool.

SuperKey® Keyboard enhancer. Simple macros turn 1000 keystrokes into 1. Also encrypts your files to keep confidential files confidential.

Borland's Electronic Reference Programs:

Turbo Lightning™ Works with all your programs and checks your spelling *while you type!* Includes 80,000-word Random House® Concise Dictionary and 50,000-word Random House Thesaurus. Forerunner of Turbo Lightning Library.

Lightning Word Wizard™ Includes ingenious crossword solver and six other word challenges. If you're into programming, Lightning Word Wizard is also a development toolbox and the technical reference manual for Turbo Lightning.

We frequently surprise people with inventive, imaginative software, and people frequently surprise us with the way *they* use it.

For example, you'll read on this page how Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, used Turbo Pascal® (and Turbo Graphix Toolbox™ and Turbo Tutor®) to cut down the tedium and time in creating Circular Performance Profile Charts (CPPCs).

We didn't know they existed, but you learn something new every day!

Applications like CPPCs might not fit your exact needs, but at the same time they might stimulate fresh ideas in your mind about how you can put Turbo Pascal and the Turbo Pascal family to work for you.

And thank you for your interest in and support for Borland International.

Philippe Kahn,
President, Borland International

INSIDE STORIES!

- Turbo Pascal 3.0, already described by *PC Magazine* as "*Language deal of the century*," is now an even better deal than that, because we've included the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95!
- Completely new Turbo Tutor 2.0 now available. New software. New manual. New split screens. New quizzes. Only $39.95. Upgrades available under Borland's "Almost-Free" upgrade plan. Details inside.

LATE NEWS!

- June/July Special Artificial Intelligence Issue of The *Micro Technical Journal* says, "*Turbo Prolog looks like it's going to be a winner, for both the beginner and professional programmer.*"

Turbo Pascal deliberately programmed to go around in circles

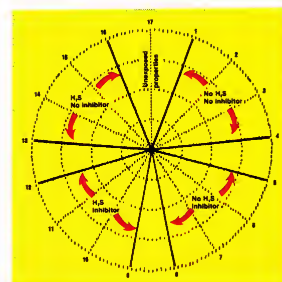
Circular charts (or CPPCs) are used by Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, to plot a single performance property for a large number of elastomers, which have elastic, rubber-like properties.

Mr. Watkins wrote us saying, "*Because CPPCs condense a lot of data in one graphic, they can be very tedious and time-consuming to draw.*"

What he did to solve those problems was to write a Turbo Pascal program for IBM® personal computers to "*generate these charts quickly and easily.*"

He used Turbo Pascal "*because it has a companion set of very powerful graphics programs (Turbo Graphix Toolbox) which greatly simplifies the required programming.*

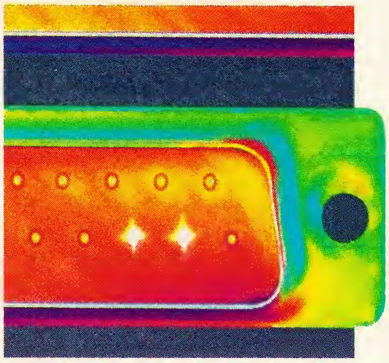
Turbo Pascal is not a difficult language to use and can be easily learned by persons who can program in FORTRAN or BASIC. An excellent tutorial (Turbo Tutor) is available for the novice or experienced programmer. The Turbo Pascal products are also very moderately priced."

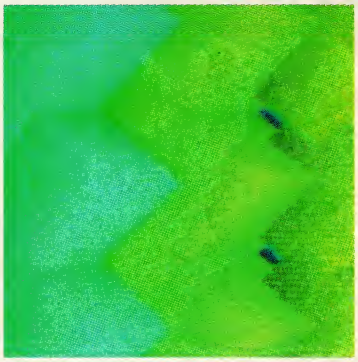


"The computer is no better than its program." Elting Morison, author of "Men, Machines and Modern Times"

PC TECH JOURNAL

SEPTEMBER 1986 VOLUME 4, NUMBER 9





Cover computer graphic · Gary Zamchick/Dadabase Design

PC TECH JOURNAL®

VOL. 4, NO. 9

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Jeff Duntemann, Caroline Halliday*
ASSOCIATE TECHNICAL EDITOR: *Dan Beale*
SENIOR COPY EDITOR: *Susan Holly*
COPY EDITOR: *Gail Shaffer*
PROOFREADERS: *Bruce Ansley, Kathleen Peddicord, Elizabeth Wardlaw*
NEW PRODUCTS EDITOR: *Carole Autenzio*
ADMINISTRATIVE ASSISTANT: *Diana Carey*
RECEPTIONIST: *JeanMarie Donlin*
CONSULTING EDITORS: *Thomas V. Hoffmann, Richard M. Foard*
CONTRIBUTING EDITORS: *Steven Armbrust, Michael Covington, Ted Forgeron, Augie Hansen, Henry F. Ledgard, Ted Mirecki, Max Stul Oppenheimer, Richard Schwartz, Robert Shostak*

ART & PRODUCTION
ART DIRECTOR: *Ina Saltz*
ASSOCIATE ART DIRECTOR: *Sharon Reuter*
ASSISTANT ART DIRECTOR: *Monique Cubicciotti*
ART SECRETARY: *Sabrina Reynolds*
ADVERTISING PRODUCTION MANAGER: *Alison Regan*
EDITORIAL PRODUCTION COORDINATOR: *James Olsson*

ADVERTISING SALES
ADVERTISING DIRECTOR: *Rita Burke*
MARKETING DIRECTOR: *Gayl Sorota*
ASSISTANT TO THE PUBLISHER: *Kathleen Abbott*
ADVERTISING COORDINATOR: *Mary Martin*
SALES SECRETARY: *Kim Schroeder*
DISTRICT MANAGERS: *Rosemarie Caruso, Arlene Braithwaite—East Coast; Pat Toohey—Mid-Atlantic; Arlene Braithwaite—Midwest; Ted Bahr, Bill Bush, Phyllis Egan—West Coast*
ACCOUNT REPRESENTATIVES: *Polly White—East Coast; Pam Sigal—West Coast; Jane Anderson—National Accounts, Mail Order*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT, Marketing: *Paul Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
PRESIDENT: *Philip B. Korsant;* EXECUTIVE VICE PRESIDENT: *James D. Dunning, Jr.;* SENIOR VICE PRESIDENTS: *Philip Sine, Kenneth H. Koppel;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, Park View, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. FAX (group 3): 301/740-8309. MCIMail: PCTECH. PCTECHline: 301/740-8383 Telex: 6502565932 MCI.

ADVERTISING OFFICES
(East Coast/Midwest) Suite 800, Park View, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing, One Park Ave., 4th floor, New York, NY 10016.

PC Tech Journal (ISSN 0738-0194) is published by Ziff-Davis Publishing Co., a division of Ziff Communications Co., One Park Ave., New York, NY 10016. Published 13 times a year; 12 monthly issues and the annual *PC Tech Journal Directory* published in November. Subscription rate is $34.97 for one year (13 issues). Additional postage for Canada and Foreign is $6.50. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to P.O. Box 2968, Boulder, CO 80321.

BPA SELECTED MARKET AUDIT DIVISION

1985 AWARD FOR BEST COMPUTER MAGAZINE Computer Press Association

Save On New Model of Periscope

"A marvelous debugging tool..." says Peter Norton.

SUMMER SALE.
You don't have to spend thousands, or even hundreds, of dollars to buy a high-performance debugger. Through September 30, the new software-only Periscope II-X is yours for only $95! Periscope II, normally $145, is only $125!

NEW MODEL.
Periscope I includes both a break-out switch for "spontaneous" debugging and a RAM board to protect the debugger from programs that overwrite memory. Periscope II includes a break-out switch that doesn't require an extra slot to install. And the new Periscope II-X is Periscope II without the break-out switch. So you can now get Periscope with no hardware at all!

NEW RELEASE.
The new version 2.1 enhances all models with many of the features users have requested. Call for details.

CHECK OUT THE REVIEWS.

- ***"Periscope strikes a balance of power and features that will please serious PC programmers."*** — Jeff Duntemann, PRODUCT OF THE MONTH, *PC Tech Journal*, 1/86
- ***"This symbolic debugger, with its breakout button and powerful command options, stands unrivaled for its flexibility . . . Periscope's diverse features, affordable price, and portability place it in a class by itself."*** —Ward Christensen, "Breaking out with Periscope," *PC Tech Journal*, 3/86
- ***"Periscope represents the finest software debugger available in its class."*** — Andrew Fried, *Computer Shopper*, 4/86

Other reviews have appeared in *Computer Language* (3/86), the Boston Computer Society's PC Report (1&2/85), and *Programmer's Journal* (Vol. 3, No. 1).

I'M CONVINCED PERISCOPE IS FOR ME!

Please send me the following:

Periscope II-X ____ @ $95* ....______

Periscope II with Switch ____ @ $125* ....______

Periscope I with Board & Switch ____ @ $295...........______

Add Shipping Cost (see below) .....______

Total Order Amount.............______

Shipping-$3 UPS; $5 COD; $8 Canada; $24 Foreign
We accept qualified PO's in the U.S.
*Prices good through September 30, 1986

Ship to:
Name ______
Company ______
Address ______
City/State/Zip ______
Telephone ______
Payment: ☐ Check enclosed ☐ COD ☐ MC ☐ Visa
Card No. ______
Expires ______
Signature ______

PERISCOPE REQUIRES an IBM PC, XT, AT, Compaq, or other close compatible; 128K RAM; DOS 2.0 or later; one disk drive; and an 80-column monitor.

Take advantage of the low sale prices while they last . . . Order your Periscope today! To order, clip the coupon, or call TOLL-FREE

800/722-7006

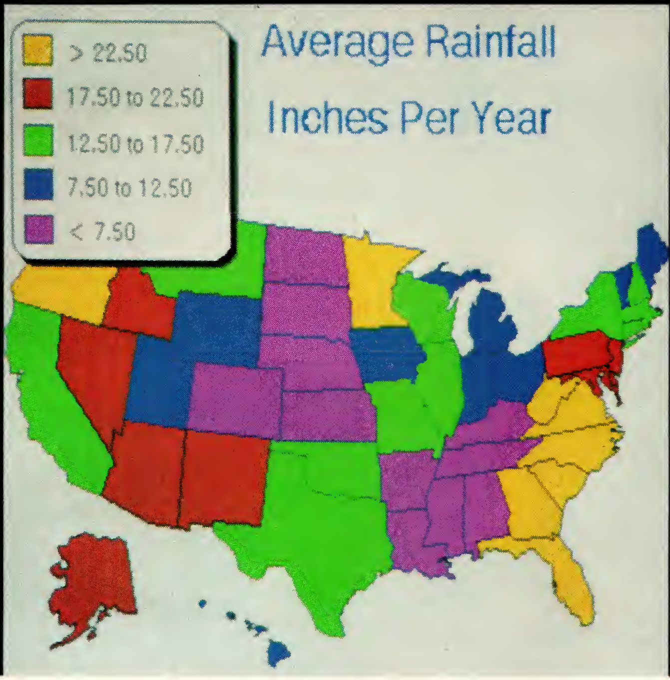
30-Day Money-Back Guarantee

Get Your Programs Up and Running;

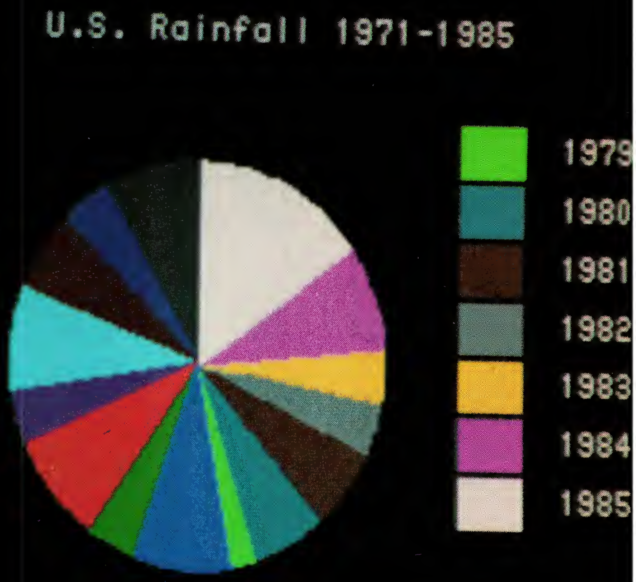
UP PERISCOPE!

Data Base Decisions
14 Bonnie Lane
Atlanta, GA 30328
404/256-3860

CIRCLE NO. 215 ON READER SERVICE CARD



EGA Mode



Plantronics Mode

Finally, An EGA Car Responds To Yo

Introducing The Paradise AutoSwitch™ EGA Card.

You've decided you want an EGA Video Card because it offers the best quality text with high resolution color graphics. The problem is that older versions of popular software, like Lotus 1A, (and some current versions too) won't work with the EGA standard. That's a problem because in the course of your workday you'll need to use software written for different video modes.

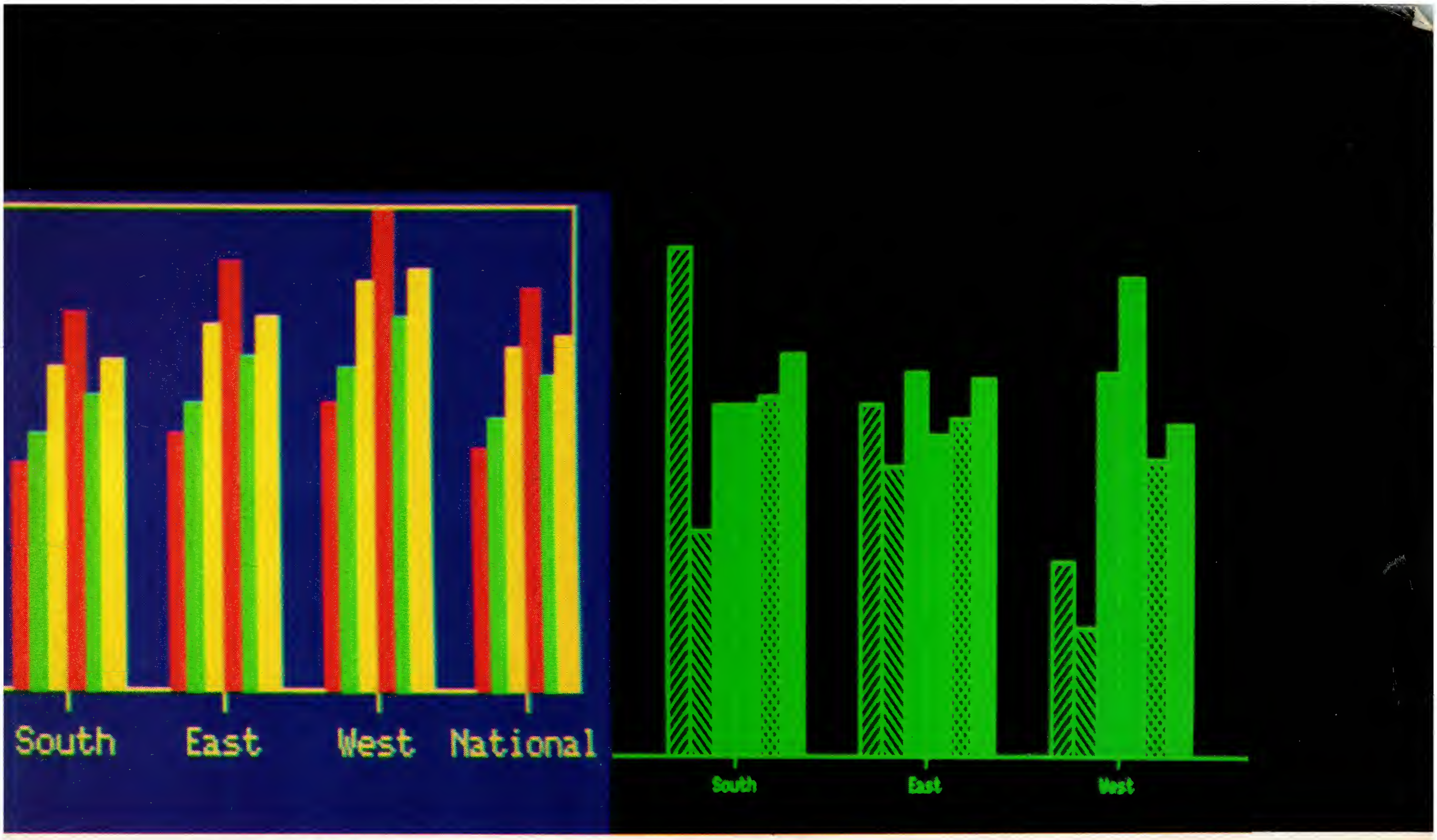
Paradise has the answer. The Paradise AutoSwitch EGA works with every popular PC video mode. And is smart enough to switch to the right mode every time you change programs. Automatically.

AutoSwitch. Only Paradise has it.

The Paradise AutoSwitch EGA changes video modes automatically to work with your software. No Mode commands. No special software to run. On RGB monitors it "autoswitches" between EGA, CGA (IBM Color Graphics) and Plantronics modes; on monochrome monitors between EGA, MDA (IBM Mono Text) and Hercules modes. The Paradise AutoSwitch EGA runs any PC software you want. Immediately. Automatically. That's something ordinary EGA Cards can't do.

Runs all EGA, CGA, MDA and Hercules Software with true hardware level compatibility.

True hardware compatibility makes the Paradise AutoSwitch EGA Card different from other EGA Cards. No other EGA Card offers the



CGA Mode

Hercules Mode

d That Automatically ur Every Mode.

6845 CRT controller included on the IBM Monochrome, IBM Color and Hercules Cards. The Paradise-designed PEGA video controller chip has the equivalent of a 6845 inside. That means when the Paradise AutoSwitch EGA runs EGA software, it's an EGA card; when it runs IBM Color Graphics (CGA) software, it's a CGA card; when it runs Hercules software, it's a Hercules card. No "batch files," special software or complicated installation. No incompatibility hassles. EGA with true 6845 hardware compatibility.

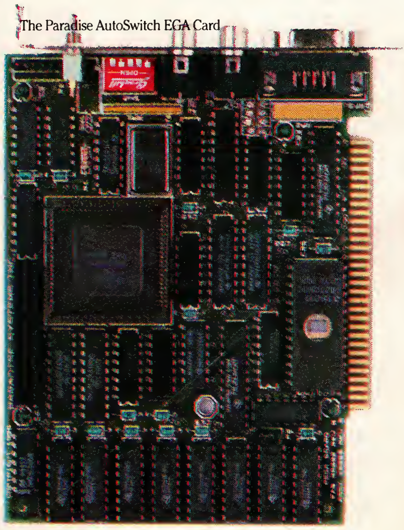
The Simpler the Better.

Paradise's exclusive AutoSwitch feature and true 6845 hardware compatibility make using the Paradise AutoSwitch EGA simple and hassle free. The Paradise-designed single chip EGA video controller makes possible all this functionality and intelligence on a short card. In fact, even the users manual is short. Only 24 pages.

The Paradise AutoSwitch EGA Card. Just what you would expect from the innovators in Video Technology.

PARADISE
SYSTEMS, INC

The Paradise AutoSwitch EGA Card



CIRCLE NO. 137 ON READER SERVICE CARD

For more information call 800-527-7977 Ext. 357, toll free. From California, 800-822-2020 Ext. 357. Or stop by your favorite computer store.

WHAT'S THE SECRET DEBUGGING WEAPON USED BY EVERYBODY FROM BORLAND TO ORACLE? FREE 44-PAGE ANSWER FROM ATRON.

"WE COULDN'T HAVE DONE IT WITHOUT ATRON'S HARDWARE-ASSISTED SOFTWARE BUGBUSTERS."

Philippe Kahn — Borland Pres.
Larry Ellison — Oracle Pres.

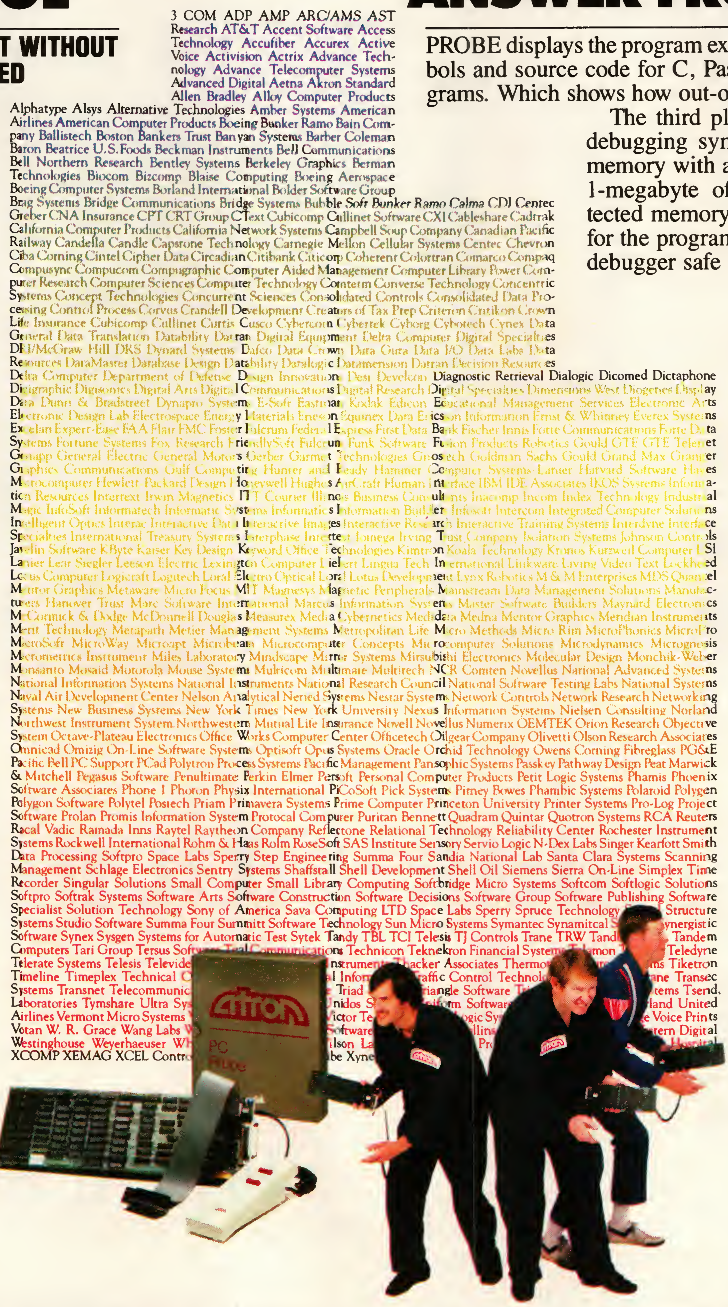
This is the city saved by the Atron bugbusters. Your city. Full of wizards, with hundreds of millions of dollars invested in wringing every ounce of intelligence and performance out of your PC. It used to be plagued with the toughest software bugs known to mankind.

PLAGUES OF BIBLICAL PROPORTIONS

The first and most difficult plague was impossible to trap with software debuggers. These were carnivorous bugs which randomly overwrote programs, data, even the debugger. Nastiest were the ones that slipped in once every few hours, or changed their behavior after each new compile. Forty days and forty nights of recompiling, *trying something else*, caused many a would-be resident of the city to run screaming into the wilderness, never to be heard from again.

Second came the plague of not knowing where the program was, or where it had recently been. This compounded the first plague: How could anyone know *what* caused the random memory overwrites? Add to this random interrupts and timing dependencies, and you begin to understand *The Fear* that gripped the city.

Then came the last plague, which brought the wizards to their knees before they even started debugging. Their towering programs consumed so much memory, there wasn't enough room for their symbol table, let alone debugging software. Even if they could get past the first two plagues, this one killed their firstborn software.



ENTER THE HARDWARE-ASSISTED SOFTWARE BUGBUSTERS

The Atron solution came as a revelation: Monitor every memory reference and every instruction executed, by adding a hardware board to the AT or PC with an umbilical probe to the processor.

The result? Wham! The PC PROBE™ and the AT PROBE™ saved civilization as we know it. The first plague was cured with PROBE'S hardware-assisted breakpoint traps on reading, writing, executing, inputting and outputting. These could be done on single or ranges of addresses, and could include particular data values. All in real time. For a mere software debugger to attempt this, a 1-minute program would take 5 hours to execute.

The second plague, not knowing from whence you came, was cured with PROBE'S real-time trace memory. The history of program execution is saved on-board, in real time. Once a hardware trap has occurred, PROBE displays the program execution in detail, including symbols and source code for C, Pascal, or assembly language programs. Which shows how out-of-range pointers got that way.

The third plague, not enough room for the debugging symbol table to be co-resident in memory with a large program, was cured with 1-megabyte of on-board, hidden, write-protected memory. System memory was then free for the program, keeping the symbol table and debugger safe from destruction.

When the job of bugbusting was done, the wizards used their PROBEs as performance analyzers. So they could have both reliability *and* performance. So they could send only the best software into the field.

IF YOU AREN'T AN ATRON CUSTOMER, ODDS ARE YOU WON'T BE MAKING THE TOP-TEN LIST.

On any given week, at least nine of the top ten best-selling software packages on the Soft-Sel Hotlist come from Atron customers.

Ever heard of Borland? "Without Atron," says its president Philippe Kahn, "there wouldn't be a Side-Kick™, Turbo Lightning™ would be light-years away, and Turbo Prolog™ wouldn't be shipping today."

Ever use a spreadsheet? From Enable™ to Paradox™, their bugs were busted by Atron products.

Into DBMSs? Everyone from Ashton-Tate to Oracle owns at least one Atron bugbuster.

If you use a product from one of the companies in *The City*, you owe life as you know it to Atron. Our guess is that 99% of all PCs, XTs and ATs have at least one product debugged with Atron bugbusters.

FREE 44-PAGE BUGBUSTING BIBLE COULD MAKE YOU A PROPHET, AND YOUR COMPANY A PROFIT.

We've written a complete tutorial on state-of-the-art bugbusting. And it's yours, free for the asking. Full of examples and illustrations, it will show you how the wizards work their magic.

If you're tired of suffering the wrath of program bugs, call Atron today. You could be busting bugs, and sales records, tomorrow.

Atron

THE BUGBUSTERS

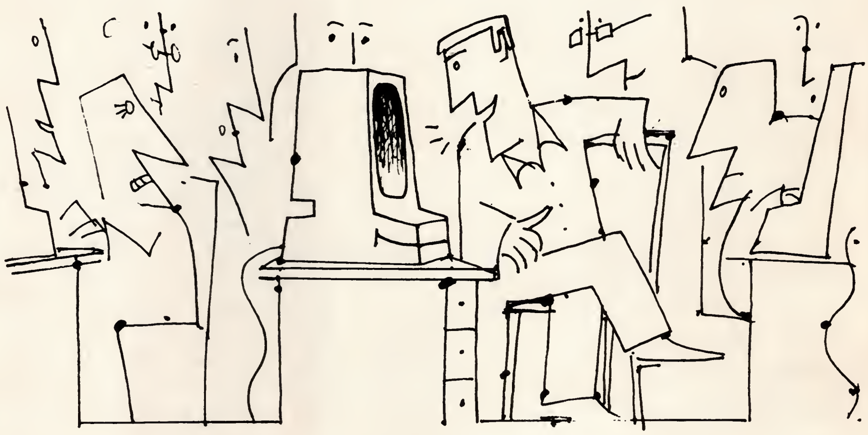
20665 Fourth Street • Saratoga, CA 95070 • 408/741-5900

Adv. by TRBA.

CIRCLE NO. 203 ON READER SERVICE CARD

Onward, FORTH

Cult, cul-de-sac, or cure?



ILLUSTRATIONS • MACIEK ALBRECHT

Have you ever tried to talk to anyone about APL? Or SNOBOL? If you play the APL zealot, nonbelievers are ready to cast you out. If you play the nonbeliever, zealots dismiss you as ignorant, imbecilic, or even worse. So few among us have a balanced view—one that recognizes the value of APL or SNOBOL for their respective capabilities. I, of course, have been blessed with just this cosmic perspective.

Balance notwithstanding, FORTH is perceived as weird, exotic, or esoteric, so I run the risk of being labeled an eccentric if I speak of the language in supportive terms. On the other hand, if I claim that FORTH is weird, exotic, and esoteric, I run the risk of having my soul damned for all eternity by the zealots. I hate having to take a position under such no-win conditions, but I happen to think FORTH has a definite place among microcomputer languages.

My FORTH investigations began pre-IBM PC. My company at that time was cramming giant doses of function into tiny microprocessors. We developed a threaded language for the Intel 4004, allowing us to write incredibly large applications considering the amount of memory we had. FORTH and our language shared many attributes.

FORTH applications are typically small in comparison to their functionality, and their performance is high. This is what has made the language so popular for dedicated microprocessor applications, especially when the chip is the Intel 8041 or 8051, which has limited amounts of ROM and RAM onboard. A particular advantage is that FORTH's kernel can be trimmed to the minimum set of words required to operate the application; no excess baggage is needed.

IBM desktops do not, however, qualify as memory-poor, resource-impaired computer systems, particularly by today's standards. The *PC Tech Journal* audience has plenty of main memory, not to mention tens of megabytes of hard disk. So just what characteristics of FORTH could make it attractive to PC software developers? The answer is that the qualities mentioned above make FORTH a good *systems language*, a high-level language tribute that is usually paid only to C.

When the PC was first announced, an associate of mine ported the LINES program, originally published in *Byte*, to the PC to demonstrate the machine's graphics capability. LINES draws a sequence of randomly sized lines; whenever an endpoint hits the edge of the screen, the line "bounces" and the sequence walks off in a new direction. Once a reasonable number of lines are on the screen, the program begins to erase previously drawn lines.

LINES originally was written in BASIC, and its performance on the PC was not very impressive. The conversion to FORTH was simple, which I considered impressive enough, but the performance improvement was dramatic—so much so that I pulled out the BASIC compiler, predicting that it would make short work of the FORTH version. Not so. The FORTH version remains one of the better performers in this admittedly weak benchmark.

Now we have size and good speed as apparent advantages. So why hasn't this language won more votes? Why isn't it the language of BIOS? Why doesn't Microsoft have a version?

IT'S ALL RPN TO ME

If you use and like HP calculators, then FORTH is already second nature. If you like TI's calculator line, and think Reverse Polish Notation is a bad dream, hatred of FORTH is easy to understand. Many programmers find FORTH's stack architecture and reverse syntax unnatural. Those accustomed to the strong structure of Pascal or C, especially the handling of subroutine parameters, do not like handling the parameters explicitly. The placement of key words in the FORTH conditional structure can be confusing, and the FORTH conventions, built up over years of use, seem as cryptic as the non-English of APL.

Actually, FORTH is no easier or harder to learn than any other language, computer or human. All languages, even BASIC, can be hideously complex under certain circumstances.

FORTH is different, just as Japanese is different. And just as Japanese is indispensible to the residents of Honshu, so can FORTH be indispensible for certain kinds of programming problems. With versions available for less than $100, it's worth a try.

APOLOGIES TO REALIA

I recently received a letter from Marc Sokol, vice president of Realia, Inc., the manufacturer of Realia COBOL. The letter is reprinted in its entirety in this month's Letters column. Mr. Sokol complained that Realia had been unfairly singled out in our review of COBOL compilers in the August 1985 issue because of its copy protection status. We labeled Realia COBOL "Not recommended" due to its copy protection, even though we would have rated it best otherwise; no other product ever has been so castigated.

He makes an embarrassingly good point. We were wrong. This was a glaring editorial oversight on our part, for which we sincerely apologize. Our conclusion from August 1985, p. 134, is amended to read "Realia COBOL (Recommended)."

To make sure *PC Tech Journal*'s policy about copy protection is clear and applied equally across products, I have instituted a new policy. Henceforth, copy protection will be identified as a characteristic of products reviewed, just as other product specifications are called out. However, no conclusions about the product will be based solely on copy protection. This is not to say that we will not editorialize about the benefits(?) or evils of copy protection, particularly as it applies to the utility or functionality of a product. What we will do is evaluate all products as though they were not copy protected and discuss copy protection as a separate issue.

Lest this be mistaken for a change in our view of copy protection, let me also make clear that we still stand foursquare against software methods of copy protection. Our reasons should be well-known to our readers by now and are also mentioned in the COBOL review. What we hope to do in the future is not condemn those products that carry copy protection, but rather to convince vendors to eliminate it. We will continue to make the case against all methods that interfere with the reasonable use of the product, and we will do so in the most constructive way we are able.

In the meantime, we are gratified that so many manufacturers have removed their copy protection schemes, especially those that make backup so difficult. We encourage others to follow suit, including Realia.

—WF

EXPERTS

This issue is the debut of a new regular department. Replacing the venerable "Legal Brief" is a column called "Expert Consultant: ..." that will focus on three topic areas.

We begin with "Expert Consultant: Applied AI," which is cowritten by Richard Schwartz and Robert Shostak, the developers of Paradox and cofounders of Ansa Software. They will write every few months on this emerging area. We are very happy to have their expertise.

The second leg is "Expert Consultant: Computer Law," by contributing editor Max Stul Oppenheimer. Max continues the excellent work he has done with "Legal Brief," which has brought both him and *PC Tech Journal* some measure of attention, particularly on issues of copy protection.

The third and final leg is "Expert Consultant: Human Factors." Henry F. Ledgard, who holds a Ph.D. from MIT, specializes in human factors research and has agreed to bring this subject to you quarterly.

This is a formidable collection of authors to keep us up to date on these complex, contemporary issues.

—WF

UNDER ONE ROOF

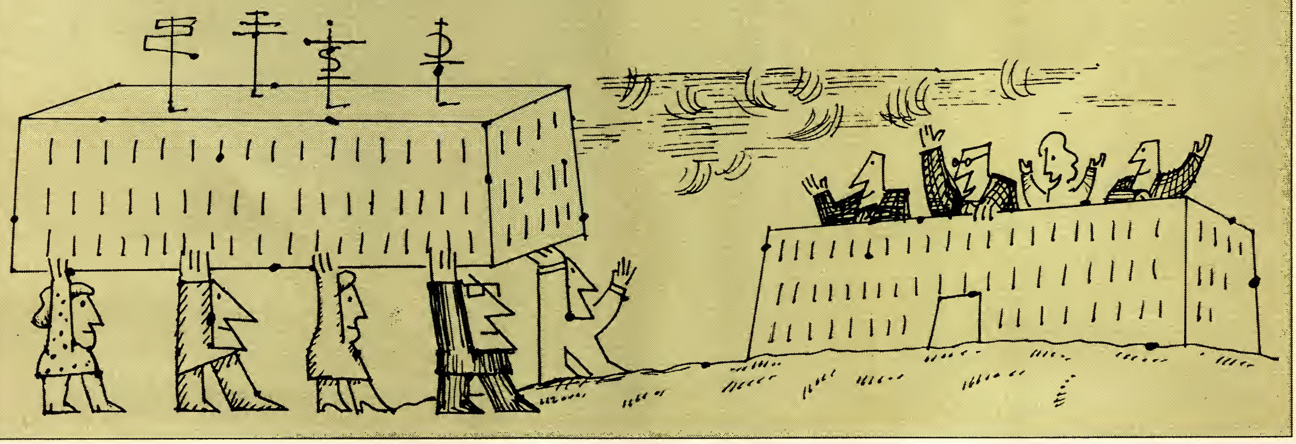
We've moved! Please use our new address and telephone number to communicate with us at *PC Tech Journal* whether you are a reader, advertiser, or author:

PC Tech Journal
Suite 800, Park View
10480 Little Patuxent Parkway
Columbia, MD 21044
301/740-8300

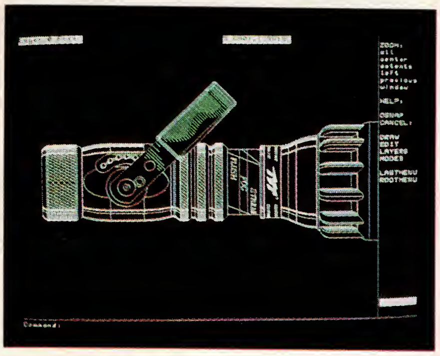
For us in editorial this is a new address in nearby, familiar territory. Our location is about 15 miles south of Baltimore and 25 north of Washington, D.C.—close enough to our old location that none of us had to change our home address to make the move. Not so for our publisher, Newt Barrett, his New York-based advertising sales staff, and our art department. They have moved from New York to Maryland because it just makes good sense for the people who work together to be together.

Will our new surroundings change *PC Tech Journal*? Absolutely. We will continue to deliver the same accurate, in-depth coverage you expect, but our ability to do so will be enhanced by a long-awaited lab, a network, a conference room, better working conditions overall, and the improved communications that inevitably result from a team of 33 people working together under one roof.

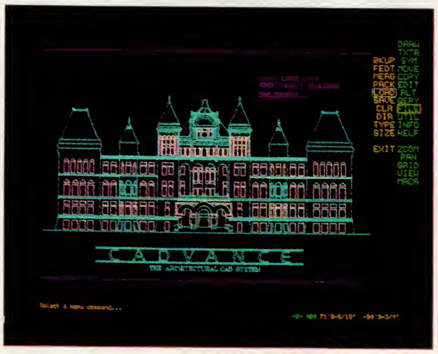
—WF



ORCHID's TurboPGA™



AUTOCAD



CADVANCE

VERSACAD

AT Owners: Professional Graphics at a ~~Reasonable~~ Affordable Price.

How We Compare:	Orchid TurboPGA	IBM PGC™	Ratio
Price	$1495	$2995	1:2
Performance			
2D 10 cm H&V lines	5000 vec/sec	1000 vec/sec	5:1
3D Cube (rotating)	320 vec/sec	82 vec/sec	3.9:1
Solids (circle)	8,380K pixels/sec	683K pixels/sec	12.3:1
Standard text (8 × 12)	14,800 chars/sec	562 chars/sec	26.3:1
Features			
Slots	one	two	1:2
Colors	256/256K	256/4K	64:1
On-board processor	80186	8088	
Flexibility			
Single Monitor System	EGA and CGA*	CGA only	
Dual Monitor System	Separate video adapter with compatible monitor required		

*Variable sync monitor and separate EGA or CGA required. EGA requires optional Orchid digital to analog EGA feature connector or Orchid EGA daughtercard.

Orchid Technology brings the high performance of a professional graphics system to the AT with the TurboPGA. Compatible with IBM's 640 × 480 resolution Professional Graphics Controller, Orchid's TurboPGA offers you more flexibility of monitors, more speed and more colors—at half of IBM's price.

With a low cost professional monitor, such as the Amdek 730, you can have an inexpensive professional two-monitor system, or with a variable-sync monitor, you can have a single-monitor graphics workstation.

This allows you to do spreadsheets, word processing, or anything else on your own video adapter and then switch to professional output for graphics.

Whether you're looking for CAD, Imaging, Industrial Controlling, Graphic Art, or Scientific Development, the TurboPGA has the colors and speed to meet the application. Since the TurboPGA is IBM compatible, it already supports such popular programs as AutoCAD, CADVANCE, VersaCAD, P-CAD, Anvil 1000MD, and MICRO CADAM.

For professionals who need high-speed, high-resolution graphics at an affordable price. Orchid's TurboPGA is the affordable solution.

47790 Westinghouse Dr.
Fremont, CA 94539
415/490-8586
Tlx: 709289

ORCHID
The Innovative Leaders

TurboPGA is a trademark of Orchid Technology. AutoCAD is a trademark of AutoDesk, Inc.; VersaCAD is a trademark of T&W Systems, Inc.; CADVANCE is a trademark of Calcomp, Inc.; IBM and IBM PGC are trademarks of International Business Machines, Inc. All other product names are trademarks of their manufacturers.

CIRCLE NO. 142 ON READER SERVICE CARD

Introducing "Wonder" Plus

The first user selectable user interface!

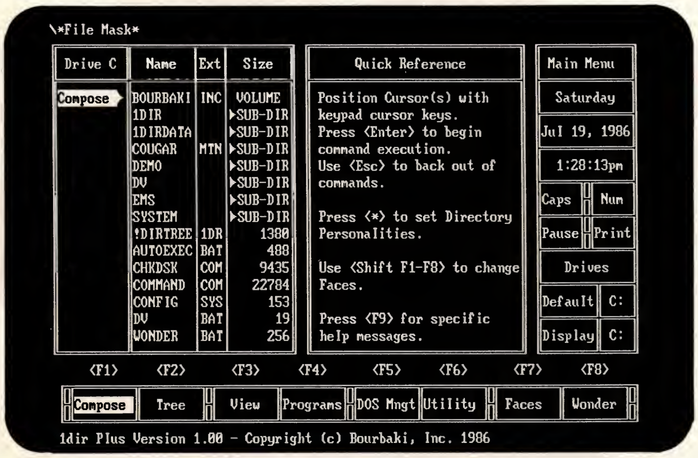
The Ultitmate Shell System for PC Coordinators, VARs, VADs . . .

The problems of different users having different needs, different applications, different "security clearances," different levels of expertise . . . have been solved.

The combination of:

FACES — (Choose from 8 screen configurations to suit your needs)
DIRECTORY PERSONALITIES — (Show only the files you want in each of your directories)
GLOBAL FILE DIRECTORY — (Perform operations across the entire hard disk)
VIEW / EDITOR — (View and edit files in ASCII, Extended ASCII and HEX - Practically a full blown word processor)
MULTI-LEVEL SECURITY — (Set up multi-user and network systems)
FILE MANAGEMENT — (Point and Shoot organization beyond the limitations of DOS)
MENU SYSTEM — (Simplify the integration of all your applications)

make *1dir* +the most POWERFUL and CUSTOMIZABLE shell system available.



QUICK REFERENCE SCREEN

Standardize without Compromise!

Never before has a single product provided the range from the simple to the sophisticated. ***1dir*** + is unique — it will not become obsolete as the user grows and gains experience. A change of Face is ALL it takes.
The BOTTOM LINE — "Have It Your Way"

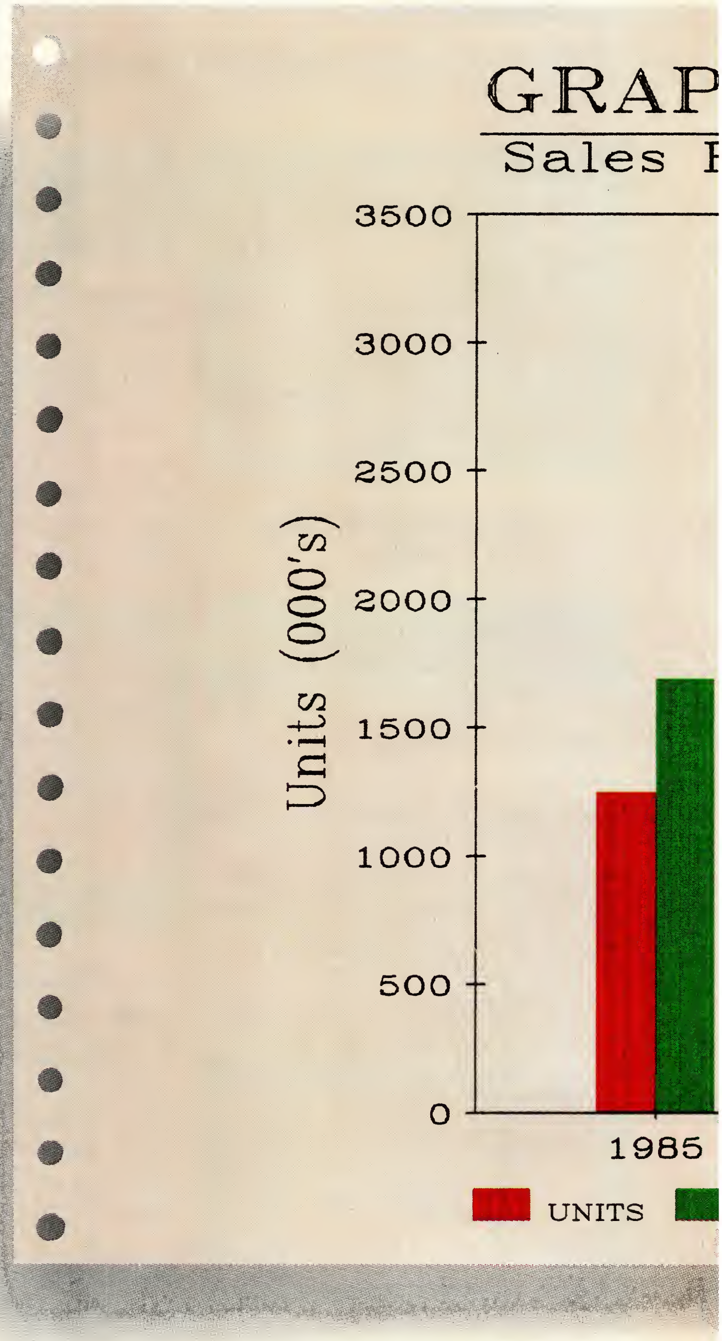
DARE TO COMPARE

1dir + combines the best features of all the existing shell programs and includes some that the competition hasn't even dreamed of.

The facts are these, the CHOICE is YOURS

Comparison Chart	*1dir* +	Commander	XTREE
Number of Screen Display Options	8	3	2
Copy, Erase, Rename, Mkdir, Move . . .	Yes	Yes	Yes
Multiple File Operations — flagging	Yes	Yes	Yes
Directory Map — Tree	Yes	No	Yes
Multi-Mode VIEW / EDITOR	Yes	No	No
Global File Directory (Entire HD)	Yes	No	No
Global File Operations	Yes	No	No
Directory Personality Capability	Yes	No	No
System Statistics Display	Yes	Yes	Yes
Change File Attributes	Yes	No	Yes
Context HELP	Yes	No	No
Multi-level Security	Yes	No	No
Locate Command	Yes	No	No
Multi-User / Network Oriented	Yes	No	No
Individual User Applications Menus	Yes	No	No
Works w/Unlimited # of Directories	Yes	?	No
Print Files	Yes	No	Yes
Print Directories and Tree Structure	Yes	No	No

NEW DASH FOR YOU



Your old printer should look so good.

Introducing the ImagEnhancer™ PC add-on board. It enables ordinary monochrome or color dot matrix printers to produce high-quality, plotter-like graphics.

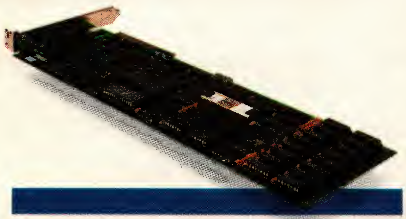
THE OUTPUT IS OUTSTANDING.

The ImagEnhancer is the perfect way to get presentation-quality graphs, charts and artwork from your old dot matrix printer.

What's more, it comes with a built-in 512K memory. So while your printer is printing one job, you can use your PC for another. Which will improve your productivity tremendously.

And of course, the ImagEnhancer works with the same computer products most everyone works with.

IBM® PCs, XTs, ATs or 100% compatibles with graphics capability. Alps, Epson® or Epson-compatible monochrome or color dot matrix printers. And Houston Instrument (DM/PL) plotter-compatible software, including Lotus® 1-2-3,® Symphony,® PFS®:Graph and Framework,™ among others.



The ImagEnhancer PC add-on board brings plotter-like output to dot matrix printers.

AND THE PRICE IS RIGHT.

At $595,* the ImagEnhancer costs a fraction of what you'd pay for a new plotter. It's simple to install, too. And backed by a one-year warranty.

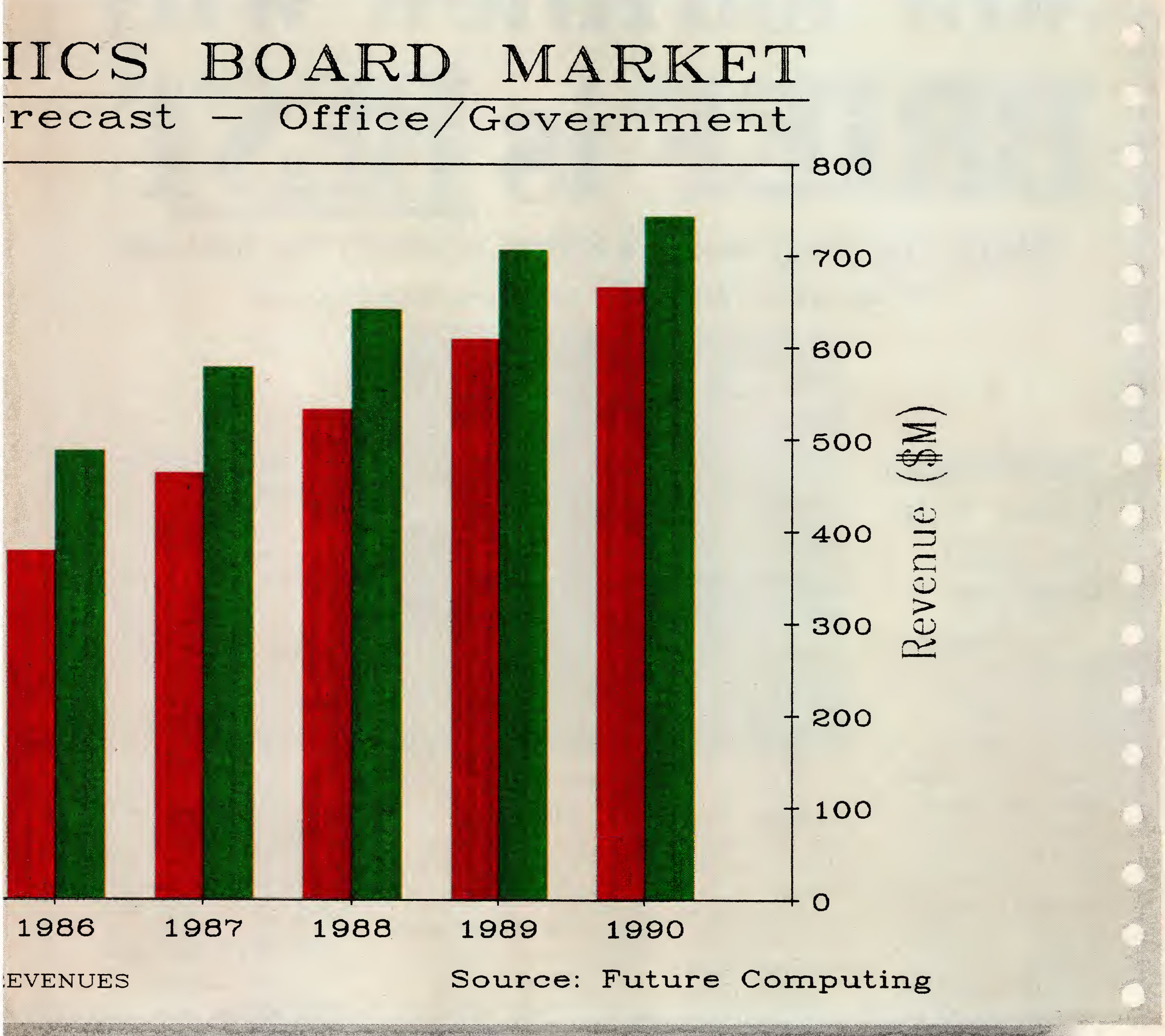
The ImagEnhancer is a product of Alps America, a U.S. subsidiary of Alps Electric Co., Ltd. of Japan, an International Fortune 500 company. Other Alps America products include a full line of high-quality printers.

For more information, print samples, or the name of the dealer/distributor nearest you, write to Alps America, 3553 North First Street, San Jose, CA 95134. Or call **(800) 828-ALPS.** In California, (800) 257-7872. In Canada, (800) 858-2577.

Better yet, send in this coupon along with your payment. And teach your old dot matrix some new tricks.

CIRCLE NO. 247 ON READER SERVICE CARD

R OLD DOT MATRIX.



This image was printed with an ordinary color dot matrix printer and the ImagEnhancer PC add-on board. With a monochrome printer, you get the same high resolution in black and white.

SO ORDER TODAY.

Name: ______________________

Address: ______________________

City: ______________________

State: __________ Zip: __________

Phone: ______________________

Send ______ ImagEnhancer(s) at $595 each.*

*Add $10.00 shipping and handling for each. California residents add 7% sales tax. Offer available in U.S.A. only.

Method of payment (circle one):

Check Money Order MASTERCARD VISA

Card number: ______________________

Exp. date: ______________________

Signature: ______________________

Mail to: Alps America
3553 North First Street
San Jose, CA 95134
Or call (800) 828-ALPS
In California, (800) 257-7872

ALPS
AMERICA

Now You Know Why BRIEF™ is BEST

"BRIEF, The Programmer's Editor, is simply the best text editor you can buy." John Dvorak, INFOWORLD 7/8/85

MACRO LANGUAGE

As Steve McMahon describes in Byte (3/85), "BRIEF is even more customizable than [another customizable editor] - not only may wholly new commands be created, but they may be assigned to any key, replacing even basic function keys like cursor control keys or the return key . . . String and integer [variable] types are available and may have either global or local scope . . . Arithmetic and logical primitives, type conversion, buffer and window control, search and translate, keyboard input, . . . are [also] available.

"Much of BRIEF was written in the BRIEF macro language, and the soruce of these macros is included with the editor. Using this source, it's possible to customize even sophisticated functions of the editor . . ."

No other editor has a more powerful macro language.

The Program Editor with the BEST Features

Since its introduction, BRIEF has been sweeping programmers off their feet. Why? Because BRIEF offers the features **MOST ASKED FOR** by professional programmers. In fact, BRIEF has just about every feature you've ever seen or imagined, including the ability to configure windows, keyboard assignments, and commands to **YOUR** preference. One reviewer (David Irwin, DATA BASED ADVISOR) put it most aptly, "(BRIEF). . . is quite simply the best code editor I have seen."

Every Feature You Can Imagine

Compare these features with your editor (or any other for that matter).

- FAST
- Full UNDO (N Times)
- Edit Multiple Large Files
- Compiler-specific support, like auto indent, syntax check, compile within BRIEF, and template editing
- Exit to DOS inside BRIEF
- Uses all Available Memory
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning Time
- Windows (Tiled and Pop-up)
- Unlimited File Size –(even 2 Meg!)
- Reconfigurable Keyboard
- Context Sensitive Help
- Search for "regular expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- A Complete Compiled Programmable and Readable Macro Language
- EGA and Large Display Support
- Adjustable line length up to 512

Program Editing YOUR Way

A typical program editor requires you to adjust your style of programming to its particular requirements – NOT SO WITH BRIEF. You can easily customize BRIEF to your way of doing things, making it a natural extension of your mind. For example, you can create ANY command and assign it to ANY key – even basic function keys such as cursor-control keys or the return key.

The Experts Agree

Reviewers at BYTE, INFOWORLD, DATA BASED ADVISOR, and DR. DOBB'S JOURNAL all came to the same conclusion – BRIEF IS BEST!

Further, of 20 top industry experts who were given BRIEF to test, 15 were so impressed they scrapped their existing editors!

NOT COPY PROTECTED

Solution Systems™

MONEY-BACK GUARANTEE
Try BRIEF ($195) for 30 days – If not satisfied get a full refund.
TO ORDER CALL (800-821-2492)

SOLUTION SYSTEMS, 335-P WASHINGTON ST., NORWELL, MA 02061, 617-659-1571

BRIEF is a trademark of UnderWare

CROSS EXAMINATION

I applaud the article on 327x emulation ("Emulating the 3278," Roger Addelson, February 1986, p. 48). For corporate mainframe environments, this is one area in which new and extant PCs are sure to be used. However, contrary to the article, the IBM 3278/79 Emulation Adapter and Control Program may run on computers other than IBMs. I have successfully tested this adapter in the Zenith 160 and 200 running IBM DOS. Because of this, I am highly suspicious of the author's assertion that the control program examines the ROM BIOS for the IBM copyright; if it looks for a copyright notice, it probably looks for the one within COMMAND.COM instead. Thus, the IBM 3278/79 Emulation Adapter may be run on any compatible ma chine that will run DOS.

John M. Scott Bryan, Ph.D.
The University of Oklahoma
Norman, OK

When we tried running the IBM 3278 Control Program on a Compaq, it would not execute. A Compaq engineer assisted us with the problem and disassembled the control program. He determined that the 3278/79 program did indeed examine ROM. With a modification to Compaq's ROM to accommodate this, the emulator worked fine.

—Roger Addelson

BASIC BUSINESS

PC Tech Journal has been a longtime favorite of BASIS employees because of its excellent coverage of new technologies. However, "Six New Shapes of BASIC" by Ted Mirecki (June 1986, p. 52) has severely tarnished the credible reputation of your publication.

BASIS must take serious exception, not only to the inaccurate details of the review of BB^x, (Business BASIC Extended), but also to the fact that you even considered including BB^x in a discussion of BASICA-compatible products.

Business BASIC is not new. It has been around longer than any of the other BASICs discussed, including Microsoft BASIC. First released in 1970 as a minicomputer operating system, it since has evolved into a serious business application dictate. Business BASIC was the only user environment available for many minicomputers—such as Basic Four, Pertec, Rexon, Quantel, and Microdata—into the 1980s.

Neither BB^x nor its true competitors attempt compatibility with BASICA, just as BASICA does not attempt compatibility with Business BASIC. They are different dialects of the language with different features and functionality that address different segments of the market. We consider it irresponsible of *PC Tech Journal* to attempt a comparison of dissimilar products.

BB^x was developed with two overriding objectives: source code compatibility with the thousands of existing Business BASIC applications and binary compatibility of application software across MS-DOS, DOS networks, XENIX, and UNIX, regardless of hardware manufacturer. While not a complete list, we have identified the following errors in Mr. Mirecki's review:

The ability to read external source code as well as to make ASCII source code available to other programs is available via the MERGE verb.

Our internal floating-point format uses 8 bytes, not 10. Whenever possible, BB^x will place the result of a computation into its integer form for more compact manipulation. No programmer assistance is required to do so.

Communications are considered a responsibility of the host system. Given that fact, BB^x will support what the host supports without interference with other applications, such as BASICA causes.

Mr. Mirecki's comments about multiple programs in a single work space are incorrect. CALLable programs are loaded externally to the work space allowing, in the large model, full use of the computer's memory.

Considerable confusion clouded the discussion of reading a line from an ASCII text file. We would suggest READ (1)A$ or INPUT (1)A$ to perform this simple operation. Mr. Mirecki seems to think that BASICA's LINE INPUT statement is more convenient.

The line that reads, "but it appears that the only way to write numeric data to a record file is to concatenate them..." is incorrect. The documentation shows that the user may read/write numerics like string data, as in:

```
WRITE(1,KEY=KEY$)VAR1$,VAR1,VAR2$,V
   AR2,ARRAY[ALL]
```

In addition to writing simple variables, BB^x also allows reading and writing entire arrays. Note that reading and writing unformatted data is done in a precisionless environment, allowing the read/write of a full precision value to a disk file without loss of significance.

Both uses of keyed files are correct in different instances. As an example, the global cross-reference utility, when it runs out of memory, will use a DIRECT file (keyed with data) to store the variable name and program names. The most typical usage is to have a reference back into a master file (say, for vendors) by various reference items (for example, by credit limit, telephone number, or name). In this case, the key would contain all of the information necessary to refer back to the original record in the master file.

The EDIT command is intended for minor changes to a single line of program code. BB^x includes this verb for syntax compatibility with earlier versions of Business BASIC. The **_edit** program does support page up (and down), as well as GOTO a specific line. These editing features are accomplished by control functions. Additionally, the editor HELP file may reside anywhere on the system, not exclusively on the A:

drive; it is normally installed in the same directory as the utilities. (Perhaps the product was not installed properly.) The review also fails to recognize that **_edit** is not simply editing a text file, but that it is actually compiling the work space program as it is being edited. This allows the programmer to break from program execution, or as the result of an error, call the editor, make the necessary changes, and then continue running the program at the breakpoint without any loss of data or program continuity.

In that our primary market is experienced developers with serious BASIC experience, we made no attempt to make the reference manual a user guide. It is designed to describe the functions and their employment.

While we will not dispute the merits of multiline functions, our greater concerns are with multilevel error trapping (for easier interaction with an operator), file and record lockout (for multiuser systems and networks), portability, screen handling, and program continuation after modification.

BB^x specifically avoids hardware-dependent functions such as PEEK and POKE. In that BB^x and its associated applications are portable from single-user PCs to UNIX systems supporting more than 100 terminals, it would be foolish for the developer to use these functions because they would lose their portability. Apparently Mr. Mirecki is unaware that even Microsoft realizes this—these functions do not exist in the XENIX version of Microsoft BASIC.

BB^x was not ported from UNIX to DOS. The development of the product for all environments was concurrent.

Mr. Mirecki failed to identify the hardware used for the benchmarks except for stating that the FILEIO.BBX was run on a floppy disk. Using an AT&T 6300 running DOS 3.1, we ran the integer and real arithmetic benchmarks almost twice as fast as in the review just by coding the program correctly.

Anyone who has doubts about the seriousness of BB^x or other Business BASICs, or the breadth and scope of the market they address, should contact a company such as Open Systems and inquire as to why it would never consider BASICA as the language of choice for serious business data processing.

Thom Olson, president
BASIS Inc.
Albuquerque, NM

I enjoy your magazine. Your reviews and programming articles are well written and I believed them to be unbiased, well researched, and responsible...until now. The review of six BASICs by Ted Mirecki has left me shocked and disappointed. His attack on BB^x by BASIS, Inc. can be described only as irresponsible, uninformed, and biased.

I have used Business BASIC on machines from the IBM PC to Basic Four minicomputers. Mr. Mirecki has not even the most rudimentary understanding of this language and its advantages over the others reviewed. His statements regarding record-oriented file access are erroneous and display his complete disregard of the manual, as does his explanation of mixed-field record handling and his ignorance of the BB^x DOS CALL feature, which lets the programmer execute any DOS function, batch file, or other program from within any BB^x program, and this within compound statements.

This article was particularly distressing because my knowledge of the product under review exceeded that of Mr. Mirecki. Moreover, he completely missed the mark not just on preference, but on documented information.

I base my business decisions on the advice in articles such as this. If a dismissal as peremptory as this one is possible, how many others have there been of which I am unaware?

James R. Knouse Sr.
Micro-Aide
Houston, TX

Mr. Mirecki's article on BASIC needs revision, especially his treatment of BB^x. As a Business BASIC programmer for more than four years, I can say that BB^x is the best implementation of BASIC I have ever used. Including BB^x in this review with PC-oriented BASICs was a poor choice. BB^x and its sister languages were designed for one specific purpose—business application programming. They have little to do with BASIC as it is generally perceived.

One of BB^x's great strengths is that it maintains its roots with older versions of the language. Programs written years ago on minicomputers have a good chance of running today under BB^x on the PC without much modification. With regard to Mr. Mirecki's review, a few errors should be corrected:

Any standard ASCII text editor can produce BB^x source code. ASCII text can be converted easily to a BB^x program file using the MERGE command. Merging from a text file to a BB^x program file is no different than running code through a compiler. Also, BB^x per-

FOUR STARS

★★★★

Paradox® is the only relational database ever to earn this honor from Software Digest.

In all, 15 relational databases were tested. And when the results were tabulated, Paradox was the only one to win high ratings in every category. From ease of learning through ease of use, error handling, performance and versatility.

So not only was it Software Digest's number one relational database, it's now the only one ever to earn their coveted four-star rating. And if you think that's good, see our new version, Paradox 1.1. For the name of the Ansa dealer nearest you, call 800-547-3000, Dept. 154; in Oregon, 503-684-3001.

PARADOX

Relational Database by Ansa Software

Software Digest Rating	Overall Evaluation	Program Name	Version Tested	Ease of Learning	Ease of Use	Error Handling	Performance	Versatility	Memory Requ	Price	
★★★★	8.1	Paradox	1.0	green	green	green	green	green	512K	$695	✓
★★★	7.8	R:base 5000	1.01	yellow	green	green	green	green	320K	$700	✓
★★★	7.3	The Smart Data Manager	2.00	yellow	yellow	green	green	green	256K	$495	✓
★★★	7.1	dBase III Plus	1.0	yellow	yellow	yellow	green	green	256K	$695	✓
★★★	7.0	PractiBase	1.14	green	green	yellow	yellow	green	256K	$70	✓
★★★	7.0	Dataease	Pre. 2.5	yellow	yellow	yellow	yellow	green	384K	$600	✓
★★	6.9	Power-base	2.10	green	green	green	red	green	320K	$349	✓

green: 7.0-9.9 yellow: 5.0-6.9 red: under 5.0

Software Digest RATINGS NEWSLETTER

Test results reprinted from cover of Software Digest March 1986.
Software Digest is a registered trademark of Software Digest, Inc., Philadelphia, Pa.

Paradox is a registered trademark of Ansa Software.

CIRCLE NO. 198 ON READER SERVICE CARD

mits the use of any editor, provided it is well behaved and does not consume all available memory, without leaving BB^x

In saving a program, the addition of PSZ after the file name automatically computes the required number of bytes necessary to save the program for the first time. Later, a SAVE will do.

Business BASIC is a record-oriented language. Because memory was at a premium in minicomputers, everything was generally written in 8KB or less. Consequently, the language has very few mechanisms that support memory management as we know it. (Large applications rely on chaining from one program to another, maintaining key variables, and rereading file. Sorts, when necessary, are written to disk as sort or keyed files.)

My BB^x manual lists six separate file types, rather than three, and, add to that a seventh, located in the verb section. ASCII text files are written STRING files using the LIST command. It is necessary in DOS to filter these files, converting >**012** to >**n**, but that is an easily automated task.

BB^x supports a formatted file structure based on the IOLIST verb. Records can be read or written using IOL=nnnn, where *nnnn* is the line number beginning the IOLIST declaration. The record fields may be any mix of strings, decimal numbers, or integers. BB^x supports floating-point operations. It is possible to write records as long concatenated ASCII strings, but that is not a standard practice.

In the last year I have worked with three implementations of interpreted Business BASIC. All of the manuals were similar in organization and defined material in much the same manner. It's great to quickly reference nine keyed files at one time, as data entry is being performed, and not index a file each time it is opened.

Lastly, BB^x portability makes it a truly outstanding product—this topic was not touched on in the review. No conversion is required when transferring either programs or data between different operating systems on which BB^x will run. Theoretically, BB^x will run on any computer that supports C.

Mel Tearle
Phoenix, AZ

The mere fact that BB^x is different from BASICA is no reason to exclude it from a comparative review of BASIC interpreters. All of the products differ to a greater or lesser extent, and one of the aims of the review was to point out such differences in a variety of products that share the major common characteristic of having the word "BASIC" emblazoned on the box. I agree with Mr. Tearle that BB^x has little to do with the typical PC-oriented BASIC, and that was precisely my warning to readers.

The title of the article, "Six New Shapes of BASIC," was not meant to imply any chronology. The fact that BB^x is not new becomes painfully apparent very quickly—its command structure is straight from the early days of microcomputing, more than a decade ago. That, and the poor documentation, are BB^x's chief failings, not its differences from BASICA or the other products.

The sketchy documentation caused me to miss some of the points made in these letters. For example, the manual's description of the MERGE command gives no clue that it reads in an ASCII file. And as Mr. Tearle points out, the files produced by the LIST command are not standard DOS text files.

Documentation problems also caused me difficulties with the file system. Specifications for READ RECORD and WRITE RECORD state that only a single string variable is allowed in these statements. I/O with plain READ and

WRITE does allow a list of variables, but insufficient information is provided on using this feature with record files. For example, record lengths must be declared in terms of bytes, but nowhere does the manual say how many bytes a numeric value takes up. Because information is given on converting numbers to fixed-length strings and on writing strings to records, I assumed that this was the only practical way of building fixed-length records.

I apologize for misstating the size of numeric variables, (with no documentation on this subject, I was guessing) and for overlooking the paging functions of the editor. But several other "errors" cited by Mr. Olson merely refer to my observations about features that are common in most BASICs but are missing from BB^x. These include communications support, multiline functions, and PEEK and POKE. Mr. Olson seems to imply that error trapping and file and record locking preclude multiline functions. And whether or not Microsoft supports PEEK and POKE under XENIX is entirely beside the point; users have come to expect it under DOS and most other microcomputer operating systems. It is quite true that using them prevents portability, but why not let users decide for themselves between portability and functionality.

As Mr. Olson concedes, the primary objective of BB^x is to support existing Business BASIC applications, and for this purpose BB^x may be worth considering. Longtime users of this dialect may become fond of BB^x, but I doubt whether many newcomers to this language would be enthusiastic about its dated design concepts. The thrust of my review was that for developing serious BASIC applications from scratch, the PC user has several better choices.

—Ted Mirecki

AUTOCAD OWNERSHIP

I ran across an attribution of AutoCAD ownership to CalComp in Victor Wright's "Mechanical CAD" (June 1986, p. 80). While a compliment to CalComp, since 1982 AutoCAD has been manufactured by AutoDesk Inc., 2320 Marinship Way, Sausalito, CA 94965.

The AutoDesk DXF file was designed as an infinitely expandable specification that could handle new enhancements to AutoCAD, and be readable in any programming language. Any computer language can read and write DXF.

DXF is now the industry standard for microcomputers, allowing other CAD programs to exchange files with AutoCAD, or with each other.

Keven A. Seaver
AutoDesk, Inc.
Sausalito, CA

PC Tech Journal *regrets the error.*

—WF

A PASCAL MISFORTUNE

We regret to announce that Software Building Blocks, Inc. is no longer in business. While the company can no longer support SBB Pascal on a full-time basis (by telephone), we will respond to written requests and reports of bugs as time permits. Send letters to P.O. Box 1293, Littleton, MA 01460.

The company will be licensing the source code of the compiler, the optimizer, the debugger, and the histogram tool that was in beta test, all for $300. Users who are interested in licensing the source should write to us for a copy of the license agreement.

We appreciate the support and apologize for any inconvenience.

Jeffrey and Laurie Moskow
Littleton, MA

We commend the Moskows for supporting the owners of an "orphaned" prod-

"Turbo Pascal... is a very good system. But don't make the mistake of trying to use it for large programs."

Niklaus Wirth
as reported in Micro Cornucopia
August-September 1985

LOGITECH

MODULA-2/86

uct. SBB Pascal was and is a good compiler; it is unfortunate that it no longer will be available.

—JD

UNPROTECTED

I read with interest "A Data Manager: The Evolving Standard" by Dave Browning (May 1986, p. 166). Among the praises for Ashton-Tate's dBASE III PLUS, Mr. Browning notes that the product is copy protected. His only negative comment about the copy protection method is that it "complicates network operating procedures." The summary page concludes, "The software is copy protected using Softguard's SuperLok, but can be installed on many fixed disks. Access modules for LAN-shared operations require a copy-protected key disk at the workstation."

No further comment about copy protection. No hesitation about recommending the product.

Please contrast this with the review in June and August 1985 ("COBOL Performs," Ted Mirecki, p. 58 and 107, respectively) of the Realia COBOL compiler. The summary read, "Not recommended. Realia is placed here reluctantly and only because of its copy protection, which, for a compiler, is totally unacceptable. The SuperLok scheme used by Realia permits running the system from a hard disk without the need for the original floppy disk, but it is not clear what it does to a hard disk when it is installed. Can it be trusted on a development system...?"

Last August, copy protection was "totally unacceptable" for a compiler (with which user applications are developed and tested). Now copy protection is quite acceptable for a database management system (with which user applications are developed and tested).

I am glad to see you are willing to recommend an excellent, copy-protected product. In the interest of fairness, please amend in print your negative conclusion about Realia COBOL.

Marc Sokol, vice president
Realia, Inc.
Chicago, IL

Mr. Sokol's letter is answered and the subject of copy protection is addressed in a sidebar included with this month's Directions column (p. 9).

—WF

ERRATUM

Please note that the volume information on the cover of the July 1986 issue should read Vol.4, No.7.

TAKE THE NEXT LOGICAL STEP

$89 Price

- Separate Compilation
- Native Code Generation
- Large Memory Model Support
- Multitasking
- Powerful Debugging Tools
- Comprehensive Module Library
- Available for the PC and the VAX

MODULA-2/86

NEW, improved Turbo Pascal to Modula-2 Translator!

Call for information about our VAX/VMS version, Site License, University Discounts, Dealer & Distributor pricing.

To place an order call our special toll free number:

800-231-7717

In California:

800-552-8885

Move up to LOGITECH MODULA-2/86. Whether you're a single programmer or part of a team, with LOGITECH MODULA-2/86 you'll decrease your overall development cycle and produce more reliable, more maintainable code. Build your program using our extensive library modules, your own modules or those from a growing list of available third-party software vendors. If you're a Turbo Pascal user you can even take your existing code along with you with the help of our new Translator!

NEW & IMPROVED!

Turbo Pascal to Modula-2 Translator $49

Now it's even easier for Turbo users to step up to Modula-2/86. Our improved Translator changes your Turbo source code into Modula-2/86 source, solving all the incompatibilities, and translating the function calls of Turbo into Modula-2/86 procedures. Implements the complete Turbo library!

LOGITECH MODULA-2/86 $89

Complete with Editor, Run Time System, Linker, 8087 Software Emulation, Binary Coded Decimal (BCD) Module, Logitech's comprehensive library, Utility to generate standard .EXE files. AND more!

- LOGITECH MODULA-2/86 with 8087 Support $129
- LOGITECH MODULA 2/86 PLUS $189

For machines with 512K of RAM. Takes advantage of larger memory to increase compilation speed by 50%.

Turbo Pascal is a registered trademark of Borland International.

RUN TIME DEBUGGER $69

(Source level!)

The ultimate professional's tool! Display source code, data, procedure call chain and raw memory. Set break points, assign values to variables, pinpoint bugs in your source.

UTILITIES PACKAGE $49

Features a post-mortem debugger (PMD). If your program crashes at run time the PMD freezes the situation so you can pinpoint, in the source, the cause of the error and the status of the data. Also includes a disassembler, cross reference utility and version that allows conditional compilation.

LIBRARY SOURCES $99

Source code for our major library modules is now available for customization or exemplification.

WINDOW PACKAGE $49

Now you can build true windowing into your Modula-2 code. Powerful, though only 15K in size. Features virtual screens, color support, overlapping windows and a variety of borders.

MAKE UTILITY $29

Automatically selects modules affected by code changes to minimize recompilation and relinking. Even figures out dependencies for you!

CROSS RUN TIME DEBUGGER AND ROM PACKAGE $199

Now available at an introductory price!

YES I want to move up to LOGITECH MODULA-2/86!

Here's the configuration I'd like:

- ☐ Logitech Modula-2/86 $89
  - ☐ with 8087 support $129
  - ☐ Plus Package $189
- ☐ Turbo to Modula Translator $49
- ☐ Run Time Debugger $69
- ☐ Utilities Package $49
- ☐ Library Sources $99
- ☐ Window Package $49
- ☐ Make Utility $29
- ☐ ROM Package $199

Total Enclosed $________

☐ Visa ☐ Mastercard ☐ Check Enclosed

Card Number ________ Expiration Date ________

Signature ________

Name ________

Address ________

City, State ________

Zip ________ Phone ________

LOGITECH

Logitech, Inc.
805 Veterans Blvd.
Redwood City, CA 94063
Telephone 415-365-9852

For European pricing please contact:

Logitech SA
Box 32, CH-1143
Apples, Switzerland
Telephone 41-21-774545

CIRCLE NO. 229 ON READER SERVICE CARD

A few good reasons

WORDPERFECT
WORDPERFECT CORPORATION
DBASE III PLUS
ASHTON-TATE
EASY BUSINESS SYSTEM PLUS SERIES
COMPUTER ASSOCIATES MICRO PRODUCTS DIVISION (SORCIM/IUS)
MICROSOFT WORD
MICROSOFT CORP.
OPEN SYSTEMS ACCOUNTING SOFTWARE
OPEN SYSTEMS, INC.
HARDISK ACCOUNTING SERIES
GREAT PLAINS SOFTWARE
R:BASE 5000
MICRORIM
NETWORK REVELATION
COSMOS
WORDSTAR 2000
MICROPRO INTERNATIONAL
MULTIMATE ADVANTAGE
MULTIMATE INTERNATIONAL CORP.
DATAFLEX
DATAACCESS CORP.
MULTIMATE PROFESSIONAL WORD PROCESSOR
MULTIMATE INTERNATIONAL CORP.
MDBS III
MICRO DATA BASE SYSTEMS, INC.
KNOWLEDGEMAN
MICRO DATA BASE SYSTEMS, INC.
SOLOMON III
TLB
MBA ACCOUNTING SOFTWARE
MICRO BUSINESS APPLICATIONS
EXCALIBUR PLUS SERIES
ARMOR SYSTEMS INCORPORATED
INFORMIX-SQL
RELATIONAL DATABASE SYSTEMS
SAMNA WORD III
SAMNA
CYMA GENERAL BUSINESS
CYMA/MCGRAW HILL
ACCOUNTING SOFTWARE
LIBRA PROGRAMMING
DBMAN-NET
VERSASOFT CORPORATION
SAMNA+
SAMNA
FASTRAK
CORE SOFTWARE, INC.
ZIM
ZANTHE
LAN: DATASTORE
SOFTWARE CONNECTIONS
INFORMA
UNLIMITED PROCESSING, INC.
INTEGRATED ACCOUNTING SOFTWARE
PROMARK LTD.
TEAM-UP
UNLIMITED PROCESSING, INC.
B.O.S.S.
AMERICAN PLANNING CORP.
METAFILE
METAFILE INFORMATION SYSTEMS, INC.
OFFICE WRITER
OFFICE SOLUTIONS
Q-PRO 4
QNE
THE BIG 8 ACCOUNTING
MODTECH
GOLDATABASE
GOLDATA COMPUTER SERVICES, INC.
THE SENSIBLE SOLUTION
O'HANLON COMPUTER SYSTEMS
MICRO-ADAPT
DATA FINANCIAL SYSTEMS
EXCALIBUR SERIES
ARMOR SYSTEMS
BBX-BUSINESS BASIC EXTENDED
BASIS INCORPORATED
CLEO 3270 BSC CLUSTER
CLEO SOFTWARE
HIGGINS ADMINISTRATIVE SOFTWARE
CONETIC SYSTEMS INCORPORATED
NETWORK COURIER
CONSUMERS SOFTWARE, INC.
HARDISK ACCOUNTING SERIES
GREAT PLAINS SOFTWARE

TOPVIEW
IBM
SMART SOFTWARE SYSTEM
INNOVATIVE SOFTWARE CORP.
MBP COBOL
MBP SOFTWARE AND SYSTEMS
WORDSTAR 2000 PLUS
MICROPRO INTERNATIONAL
CC:MAIL
PPC/SYSTEMS
ACCOUNTS PAYABLE
REALWORLD CORP.
GENERAL LEDGER
REALWORLD CORP.
JOB COST
REALWORLD CORP.
INVENTORY CONTROL
REALWORLD CORP.
RM/COBOL
RYAN-MCFARLAND CORPORATION
BTRIEVE/N
SOFTCRAFT, INC.
OPEN ACCESS 2
SOFTWARE PRODUCTS INTERNATIONAL, INC.
DBIII COMPILER
WORDTECH SYSTEMS, INC.
MARKET MANAGEMENT SYSTEM
ABACUS
PRISMA PRODUCTION
ABC SYSTEMS AG
C-MAIL/C-MERGE
ACCESS DYNAMICS, INC.
CENTRAL SERVICES MGMT SOLUTION
ADVANCED HEALTHCARE SYSTEMS, INC.
HOT PRINT
ADVANCED PC SOLUTIONS
MARCON II
AIRS, INC.
AMBER
AMBER SOFTWARE CORPORATION
P.O. WRITER
AMERICAN TECH, INC.
COLLECT
ANOTHER DIRECTION DATA
HPMS
ANOTHER DIRECTION DATA
H.I.S.
ANOTHER DIRECTION DATA
RAISE
ANOTHER DIRECTION DATA
PHYSICIANS BILLING
ANOTHER DIRECTION DATA
CUMS
ANOTHER DIRECTION DATA
PROJECT WORKBENCH
APPLIED BUSINESS TECHNOLOGY
TELECONFERENCING
APPLIED COMPUTER RESOURCE SERVICES
JOB TRACKING
APPLIED COMPUTER RESOURCE SERVICES
TEN-IN-ONE
ARMOR SYSTEMS
MAINSTREET MERCHANDISER
ARMOR SYSTEMS
DONOR TRACKING SYSTEM
ARMSTRONG BUSINESS COMPUTERS
DENTALIS SYSTEM VI
ARTICULATE PUBLICATIONS, INC.
MEDCALIS SYSTEM VI
ARTICULATE PUBLICATIONS, INC.
PM/TURBO
ASPEN COMPUTER SYSTEMS, INC.
DEX
ATLANTA TECHNOLOGIES
CALL-CONTROL
ATLAS SYSTEMS
SERVICE MASTER
AVNET COMPUTER TECHNOLOGY
AUTOMATED PROOF CORRECTION SYSTEM
BACKROOM SYSTEMS GROUP
RESEARCH & ADJUSTMENT CONTROL
BACKROOM SYSTEMS GROUP
LARGE ITEMS MONITORING SYSTEM
BACKROOM SYSTEMS GROUP
RETURN ITEMS CONTROL SYSTEM
BACKROOM SYSTEMS GROUP

AUCTIONEER 3 & LISTRIEVE
BAROQUE SOFTWARE SYSTEMS
TRACTOR & IMPLEMENT DEALERS ACCOUNTING
BASIC SOFTWARE SYSTEMS
PC NETWORKING PRODUCTS
BRAEGEN CORP.
NETUTILS
BRITISH RAILWAYS BOARD
TIME ANALYSIS AND BILLING
BUSINESS APPLICATIONS SOFTWARE, INC.
CLIENT ACCOUNTING/GL
BUSINESS APPLICATIONS SOFTWARE, INC.
THE OFFICER
BUSINESS SYSTEMS RESOURCES
PM/TURBO
CALDWELL COMPUTER ENTERPRISES
CHARTERHOUSE ACCOUNTING SYSTEMS
CHARTERHOUSE SOFTWARE CORP.
CHARTER ACCOUNTING SYSTEMS
CHARTERHOUSE SOFTWARE CORP.
CB86 TO NOVELL INTERFACE
CHECKPOINT SYSTEMS
VIDWARE
CHECKPOINT SYSTEMS
AUTOBOOK
CISCO, INC.
WHEELS
CISCO, INC.
CITATIONS
CISCO, INC.
TIES
CISCO, INC.
CAD
CISCO, INC.
IMIS
CISCO, INC.
CASS
CISCO, INC.
POSSE
CISCO, INC.
ELECTRONIC PEGBOARD
COLONIAL VALLEY SOFTWARE, INC.
NOVELLIB & NOVELLIB-L
COMMUNICATION HORIZONS
PIER
COMP-TRON, INC.
EARTHWK
COMP-TRON, INC.
BEAMS
COMP-TRON, INC.
CTCOGO
COMP-TRON, INC.
STABY
COMP-TRON, INC.
PGELEV
COMP-TRON, INC.
STRAN
COMP-TRON, INC.
CADLINK
COMP-TRON, INC.
POLICY EXPRESS
COMPARE DATA CORPORATION
COMPUNEWS CLASSIFIED AD MANAGEMENT
COMPUNEWS, INC.
SCRIPMASTER/PC
COMPUTER, INC.
THE PROGRAM MANAGER
COMPUTERIZED BUSINESS SOLUTIONS
HERITAGE LIVESTOCK AUCTION
COMPUTERIZED SERVICE AND DESIGN
MENU SYSTEM
COTTAGE INDUSTRIES, INC.
4GL
CRAIGMORE HOUSE, LTD.
CYBERGRAM WIDE AREA MAIL
CYBERTEK COMPUTER PRODUCTS, INC.
MICRO INFORMATION MANAGER
CYBERTEK COMPUTER PRODUCTS, INC.
CYBERSCRIBE V
CYBERTEK COMPUTER PRODUCTS, INC.
AUTOSERV
CYLLON SYSTEMS INTERNATIONAL, LTD.

MAILING LIST
DATABASE GROUP
PROF TIME MANAGER AND BILLING
DATAFLOW
THE TERRA SYSTEM
DATASPHERE COMPUTER SYSTEMS, INC.
CLIPLOCK
DELTA CONTRACTING SERVICES, INC.
DS BACK UP
DESIGN SYSTEMS, INC.
MAIL ORDER SYSTEM
DIRECT BUSINESS SOLUTIONS
PSI/RIS,MOS/IS,MED/SIS
DIVERSIFIED BUSINESS SYSTEMS, INC.
RODE/PC DATA ENTRY SYSTEM
DPX
CUSTOM SOFTWARE
EAGLE ENTERPRISES
ATTACH/X.25 PROGRAM
EICON TECHNOLOGY CORPORATION
ACCESS/X.25 PROGRAM
EICON TECHNOLOGY CORPORATION
TRUCKING ACCOUNTING
ELROM INTERNATIONAL, INC.
COUNTRY CLUB MEMBER BILLING
ERNEST A. JONSON AND CO.
Y-MANAGER
EXECUTIVE COMPUTER TECHNOLOGIES
OMECOM
FAR WEST SYSTEMS
LOAN PROCESSING SYSTEM
FINANCIAL INDUSTRY COMPUTER SYSTEMS
MORTGAGE LOAN SERVICING SYSTEM
FINANCIAL INDUSTRY COMPUTER SYSTEMS
COMMERCIAL/CONST. LOAN SERVICE
FINANCIAL INDUSTRY COMPUTER SYSTEMS
2001 ACCOUNTING SYSTEM
FINANCIAL INFORMATION SYSTEMS
PHARMACY MANAGEMENT SYSTEM
FOUNDATION SYSTEMS
FOURTH SHIFT CORP.
FOURTH SHIFT CORP.
CPA-11
FRANKLEN COMPUTER SYSTEMS, INC.
TCB-11
FRANKLEN COMPUTER SYSTEMS, INC.
PMS-11
FRANKLEN COMPUTER SYSTEMS, INC.
GARPAC APPAREL MGMT SYSTEM
GARPAC CORPORATION
PLANT EQUIPMENT MANAGEMENT
GENERAL PHYSICS CORPORATION
SALES PROSPECT TRACKING
GOLDATA COMPUTER SERVICES, INC.
YOUR MOVE
GOLDATA COMPUTER SERVICES, INC.
SOS
GOLDATA COMPUTER SERVICES, INC.
BASIS
GOLDCOAST SOFTWARE SYSTEMS, INC.
THE DISTRIBUTOR
HIGH TECH INDUSTRIES
RESOURCE MANAGER
HIGH TECH INDUSTRIES
CITY MANAGER
HIGH TECH INDUSTRIES
HIMED
HISOFT COMPUTERS PTY., LTD.
HIPORTFOLIO
HISOFT COMPUTERS PTY., LTD.
FAMOUS
HOLM-DIETZ COMPUTER SYSTEMS, INC.
BASIC 3.2
IBM
INFOTEAM ACCOUNTANT
INFO DESIGNS, INC.
HOME CARE INFORMATION SYSTEMS
INFOMED
BCS SOFTWARE
INFORMATIC SYSTEMS

CIRCLE NO. 112 ON READER SERVICE CARD

MICROSOFT LANGUAGES NEWSLETTER Vol. 1, No. 9

News about the Microsoft Language Family

Adding New Commands to Microsoft® QuickBASIC Version 2.00

The standard way to add separately compiled modules to a BASIC program has been through the use of a linker, but Microsoft QuickBASIC provides a second method as well. The Microsoft QuickBASIC BuildLib routine can be used to add separately compiled modules to the BASIC runtime file. You can create a new runtime file that includes software interrupt support by typing:

```
BUILDLIB USERLIB.OBJ, QB_BIOS.EXE;
```

This will create a new runtime file by the name QB_BIOS.EXE that can be used by Microsoft QuickBASIC. When you load Microsoft QuickBASIC, you must specify the new runtime file with the /L switch as follows:

```
QB/L QB_BIOS.EXE
```

This allows Microsoft QuickBASIC to support separately compiled modules without requiring the LINK step.

Using the mouse in Microsoft QuickBASIC Version 2.00 Programs

The software interrupt routines that you just added to Microsoft QuickBASIC can be used to access the ROM BIOS and DOS system service routines. These routines provide a range of mouse, keyboard, video, and DOS services that were previously unavailable to BASIC programs. The following example lets you draw in high resolution graphics mode with the mouse. The mouse driver (MOUSE. COM) must be installed before running this program.

```
DEFINT a-z                                          'everything is integer
      SCREEN 2                                      'high resolution graphics
      m0 = 0: m1 = 0: m2 = 0: m3 = 0: oldx = 0: oldy = 0
      CALL mouse( m0, m1, m2, m3 )                  'initialize mouse
      m0 = 1
      CALL mouse( m0, m1, m2, m3 )                  'turn on mouse cursor
      m0 = 3
      WHILE inkey$ = ""                             'exit when a key is pressed
            CALL mouse( m0, m1, m2, m3 )            'read mouse position
            IF m1 <> 0 THEN                         'if mouse button pressed
              LINE(oldx,oldy)-( m2, m3 )            'draw a line to mouse position
              oldx = m2: oldy = m3                  'save old position
            END IF
      WEND
END
'     Call the mouse interrupt driver
SUB mouse( m0, m1, m2, m3 ) STATIC
DIM regs( 7 )
      regs( 0 ) = m0: regs( 1 ) = m1: regs( 2 ) = m2: regs( 3 ) = m3
      CALL INT86( 51, VARPTR( regs( 0 )), VARPTR( regs( 0 )))
      m0 = regs( 0 ): m1 = regs( 1 ): m2 = regs( 2 ): m3 = regs( 3 )
END SUB
```

In addition to INT86, the userlib.obj module also contains the INT86X and PTR86 routines. The INT86X routine should be used instead of INT86 when the system call requires segment registers. The PTR86 routine is used to convert the address of a large numeric array variable to a segment-offset value for use with INT86X. Access to the software interrupts adds tremendous flexibility to BASIC programs. However, care should be taken since BASIC does no error checking on interrupt calls.

For more information on the products and features discussed in the newsletter,
write to: Microsoft Languages Newsletter
16011 NE 36th Way, Box 97017, Redmond, WA 98073-9717
Or phone:
(800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Latest DOS Versions:

Microsoft C Compiler	4.00
Microsoft COBOL	2.10
Microsoft FORTRAN	3.31
Microsoft Macro Assembler	4.00
Microsoft Pascal	3.31
Microsoft QuickBASIC	2.00

Look for the Microsoft Languages Newsletter every month in this publication.

A Slotless Clock

dClock incorporates a battery-operated clock/calendar into the PC or PC/XT without occupying an expansion slot.

IBM has a way of turning a segment of the PC market inside out with small changes in its product line. Recently, the company made some minor changes to the memory-addressing logic in the motherboard of the PC/XT. This enabled the use of four rows of RAM chips on the XT motherboard: two rows of 64KB chips and two rows of 256KB chips. With 640KB of RAM on the motherboard, further memory expansion to run high-end applications becomes an option rather than a necessity.

This puts dealers and system integrators in a difficult situation: the raison d'etre for multifunction boards has been memory expansion. Additional ports are not always required; the IBM Monochrome Display Adapter includes a parallel port, and internal modems do not require serial ports. The only other *essential* occupant of the traditional multifunction board is the battery-operated clock/calendar. Filling a precious expansion slot with empty RAM sockets and unnecessary ports to obtain a system clock may limit other expansion options. If a clock is all that is needed, a solo clock should be provided somehow—without using an entire expansion slot in the process.

Microsync, Inc. offers what may be the most elegant solution to this problem: a small (1¼-by-3¾-inch) circuit board that plugs into the CPU socket. The board incorporates only a clock and costs $60. For providing this elegant solution, dClock has earned its place as *PC Tech Journal's* Product of the Month for September 1986.

Simplicity is the watchword here. dClock contains only six chips: one is the Epson 58321 clock/calendar device; the others are I/O address decoding and bus interface devices. Nearly half of the circuit board is taken up by a 40-pin chip socket. dClock is installed in place of the 8088 CPU, which must be removed from the motherboard and installed in the socket. An additional 40-pin socket is provided as a spacer to fit between dClock and the motherboard in case some component on the motherboard interferes with dClock's position in the 8088 socket.

Users probably can install dClock in any 8088-based PC compatible machine, including computers such as the Victor 9000 that are not considered especially PC compatible. Problems encountered when trying to install dClock are likely to be mechanical rather than electrical in nature, because none of the CPU lines is interrupted; in fact, the pins of the 8088 socket on dClock protrude through the circuit board and become the header pins that plug into the 8088 socket on the motherboard. Perhaps the most difficult procedure involved with installing dClock is removing the 8088 from the PC. Most vendors of accelerator board products provide a simple chip puller; certainly such a tool could be offered with the software disk.

dClock communicates with the 8088 through I/O address 6C0. The only known I/O address conflict is with Central Point Software's Copy II PC Option Board, which can be rejumpered to a different address.

To keep costs down, driver software for dClock is provided on paper, in the form of two BASIC programs that contain the software in DATA statements. When these programs are keyed in and run, they generate two .COM files: one to set the clock and the other to read it. To verify correct keyboard entry, the BASIC programs generate checksums for each line of DATA statements; these can be compared to the correct checksums provided in the installation manual. A diskette containing both of the programs can be obtained for an additional $10. The diskette also contains an installable device driver that is invoked from CONFIG.SYS.

The device driver is far superior to the two .COM files, because it sets the clock/calendar chip automatically whenever a new time or date is entered explicitly via the DOS TIME and DATE commands. In addition, it does not require the user to make any changes to the AUTOEXEC.BAT file.

The documentation consists of 30 standard slipcase pages that provide clear, unambiguous instructions for installation of both the hardware and the software. No mention is made of the I/O address requirements, nor is information provided that would enable a system integrator to write a driver for an operating system other than DOS. The device driver should be present in paper form as well as on the optional diskette. Entering the 700 bytes of DATA statements necessary for the device driver would be a more difficult procedure (the .COM programs are scarcely 100 bytes apiece) but purchasers at least should have the option. Alternatively, this would be an ideal application for a Cauzin Softstrip (see "The Softstrip System," Product of the Month, Jeff Duntemann, May 1986, p. 29.)

The third-party PC enhancement board industry eventually will adapt to IBM's XT motherboards with 640KB. Clocks are appearing already in video boards, floppy disk controller boards, accelerator boards, and the latest generation of multifunction boards that contain megabytes of expanded memory rather than the traditional 384KB RAM for filling out 256KB motherboards. Expanding the XT is not as simple as it once was. Dealers and system integrators will need to study new solutions as they appear, and they would do well to remember that the simplest solutions often are the best.

dClock: software on paper, $59.95;
software on disk, $69.95
Microsync, Inc.
P.O. Box 116302
Carrollton, TX 75011
214/492-5265
CIRCLE 352 ON READER SERVICE CARD

Turn Your System/3X Into The Perfect Host With PCOX Technology.

PCOX™ 5250 products make your System/3X treat your PCs like members of the family.

Your System/3X and your PCs already live together. Now they can work together, too. Thanks to PCOX Technology.

With PCOX 5250 connections, your PCs enjoy the full privileges of a 5251 Model 11, 5291 or 5292.

Which means your PCs can access and transfer files from your System/3X data base, use its host as a departmental processor, or participate in your company's distributed SNA network.

MORE WAYS TO SAY HELLO. The PCOX 5250 series comes in twinax and remote versions—one for local connections, and one for connections over phone lines.

PCOX 5250 products come in twinax and remote versions, so PCs can enjoy S/3X connections in person or over phone lines.

And both versions support up to seven concurrent host sessions. No competing product delivers more.

So let PCOX Technology open doors between your System/3X and your PCs.

Call CXI today, toll-free.

800-225-PCOX.

In California, call 415-424-0700.

CXI

CXI, Inc., 3606 West Bayshore Road
Palo Alto, CA 94303. Telex: 821945

PCOX and all PCOX products are trademarks of CXI, Inc.

CIRCLE NO. 217 ON READER SERVICE CARD

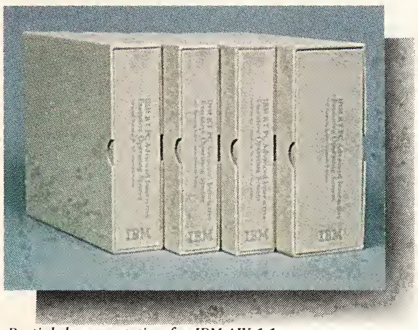
Hardware, software, and other developments for the IBM PC family

Intel's iPAT Performance Analysis Tool

FROM IBM

IBM Corporation has announced a desktop **Series/1 5170 model 496** that features a Series/1 microprocessor with 1MB of memory and a six-port terminal attachment card integrated within a PC/AT model 339. The 8-MHz AT serves as the I/O controller for the Series/1. Most Series/1 applications will run on the new model, which supports eight IBM ASCII display stations and two RS232-C communications ports. When not running as a Series/1, the computer operates as an AT. $10,695.

The company also has released **version 1.1** of the **AIX** operating system for the RT/PC. This version features shared libraries and new routines for the C compiler that produce more efficient object code, and it includes a set of graphics device drivers. Version 1.1 has an advanced display graphics library and supports the RT transmission control protocol/internet protocol (TCP/IP), which enables RTs to attach to an inter-



Partial documentation for IBM AIX 1.1

net network. Included with AIX 1.1 is version 1.1 of the Virtual Resource Manager, which supports the RT/PC baseband adapter, the IBM 6154 advanced color graphics display, the IBM 6155 extended monochrome graphics display, the lighted program function keyboard and dials, and the IBM 5083 tablet models 11 and 12. $3,400.

IBM has announced several other products for the RT: **RT/PC Data Management Services 1.1** includes a utility to check files, $500; **RT/PC SQL/RT Data Base 1.1** includes service updates, $1,000; **RT/PC/AT Coprocessor Services 1.1** offers support for the IBM 6154 advanced color graphics display and the IBM 6155 extended monochrome graphics display, $550; **RT/PC 3278/79 Emulation 1.1** includes a new program that enhances file-transfer facilities, $500; **RT/PC INmail/INet 1.1** includes service updates, $625; **RT/PC FORTRAN 77 1.1** includes a global optimizer and true single-precision, floating-point processing, $650; **RT/PC Pascal 1.1** includes service updates, $1,000; **Workstation Publishing Software 2.0.9** includes support for the IBM 6155 display and the ability to use plot files conforming to a subset of the CalComp 925 Interface, $1,995.

The **IBM RT/PC baseband adapter card** connects RTs to EtherNet LANs. When used with the TCP/IP program, users can share files, applications, and printers. The **IBM Token-Ring Network RT/PC adapter card** permits all RT models to be connected to the IBM Token-Ring Network. The adapter contains a microprocessor with code that allows users to write application programs for mail service, file transfer, and networking. Baseband adapter card, $850; Token-Ring card, $1,095.

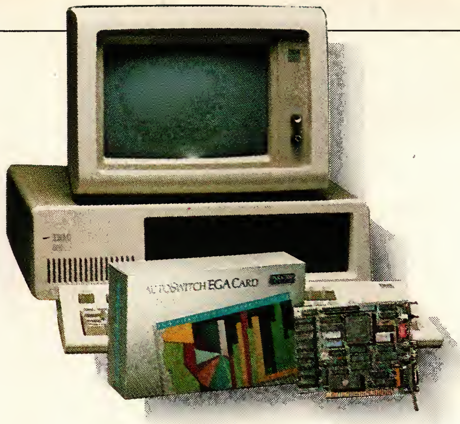
The **multiprotocol communications adapter card**, which can be connected to all RT models, provides flexible telecommunications capability. The adapter lets a variety of devices communicate with RTs remotely over telephone lines through modems using asynchronous, binary synchronous, synchronous data link control, high-level data link control, and X.21-type protocols. $850.
IBM Corporation, 900 King Street, Rye Brook, NY 10573; Contact the local IBM dealer, 800/426-2468
CIRCLE 301 ON READER SERVICE CARD

HARDWARE

Intel Corporation has introduced a high-performance, single-chip graphics coprocessor called the **82786**. The 82786 supports two independent, on-chip processors for manipulating graphics and text while executing multiple windows; it operates independently of the host CPU. This chip uses the same advanced CHMOS-III process as the 80386 microprocessor. It has 130,000 transistors, combines high performance with low power consumption, operates at less than 1 watt, and supports 32 dynamic RAM devices in its dedicated graphics memory without additional memory-support logic. To support the memory required by individual bit maps, the 82786 uses a linear, 4MB graphics memory. Within this memory, bit maps can range up to 32,000 pixels on a side, with individual pixel depth (bits/pixel) ranging from 1 to 8 bits of color or gray scale. In addition, address space can be mapped into graphics and system memory. The 82786 is less than $100 in quantities of 1,000.
Intel Corporation, Literature Department, W-300, 3065 Bowers Avenue, Santa Clara, CA 95051; 408/987-8080
CIRCLE 302 ON READER SERVICE CARD

Also from **Intel Corporation** comes an application and realtime software analysis tool hosted on the PC/XT, PC/AT, and Intel development systems. The **iPAT** (Performance Analysis Tool) consists of a base unit, a power supply, a target interface, and control software. It supports operation with Intel's I2ICE (integrated in-circuit emulator) for analysis of software based on 8086/88, 80186/286, and 80188 microprocessors. iPAT automatically uses the high-level language symbols generated by Intel assemblers and compilers, such as assembly language, C, PL/M, Pascal, Ada, and FORTRAN. These symbols can be used

MST Model 7350 Desktop Copier for diskette conversion



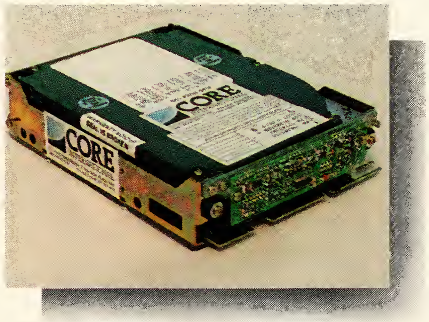
From Paradise Systems

on iPAT to quantify code behavior at the module, procedure, statement, or address-range level. iPAT monitors the activity of software generated with Intel compilers, specifying and displaying the actual program module names, procedure names, or line numbers. Code from other compilers and assemblers can be analyzed using absolute address ranges or by creating command files that associate program symbols with address ranges. Up to 125 ranges can be profiled simultaneously. During code coverage mode, 252KB of object code can be analyzed simultaneously. $9,995.
Intel Corporation, Literature Department, W-303, 3065 Bowers Avenue, Santa Clara, CA 95051; 408/987-8080
CIRCLE 303 ON READER SERVICE CARD

The **AutoSwitch EGA Card**, a new video display adapter for the PC family has been introduced by **Paradise Systems, Inc.** The AutoSwitch EGA automatically switches between the IBM EGA mode and all other popular video display modes. This short card includes 256KB of video memory and provides flicker-free scrolling. The card is based on a proprietary application-specific semiconductor device, the PEGA1 video controller chip, that supports all previous IBM video display standards by providing the registers for a Motorola 6845. By embedding the 6845 chip in the PEGA1 chip, all DOS software products written to the MDA, CGA, Hercules, and Plantronics on ColorPlus will run on the AutoSwitch EGA Card. $599.
Paradise Systems, Inc., 217 E. Grand Avenue, San Francisco, CA 94080; 415/588-6000
CIRCLE 311 ON READER SERVICE CARD

A new hardware/software system for scientific digitizing and measurement has been introduced by **Jandel Scientific**. The **Sigma-Scan** system features a high-resolution digitizing tablet and software for making a variety of two-dimensional measurements. Sigma-Scan can measure areas, distances, perimeters, lengths of curving lines, angles, and slopes, in addition to its object counting and single-point, incremental and stream x,y digitizing abilities. Standards for measurement can be set by the user. The program also can perform descriptive statistical analysis on the obtained data, including means, standard deviation, correlations, t-tests, transforms, linear regressions, and screen plots. Data can be saved to disk on standard ASCII or DIF files and can be loaded into other programs, such as Lotus 1-2-3, SIGMA-PLOT, and CRISP. $1,195, software-only version, $395.
Jandel Scientific, 2656 Bridgeway, Sausalito, CA 94965; 415/331-3022
CIRCLE 310 ON READER SERVICE CARD

A **half-height drive** available in both 26MB and 43MB capacities has been announced by **CORE International**. Both



A new half-height drive from CORE International

versions feature automatic park and lock, full shock mounting, dedicated servo surface, voice coil technology, low power consumption, and low heat generation. The average access time of both drives is 26 milliseconds. The drives support DOS and XENIX. 26MB model, $1,395; 43MB model, $1,695.
CORE International, 7171 N. Federal Highway, Boca Raton, FL 33431; 305/997-6055
CIRCLE 306 ON READER SERVICE CARD

A stand-alone system for converting 5¼-inch diskette software to run on 3½-inch diskettes has been introduced by **Media Systems Technology, Inc.** The **Model 7350 Desktop Copier** converts and duplicates both 5¼-inch and 3½-inch programs from Ashton-Tate, Lotus Development Corporation, and Microsoft. Software for the PC Convertible also can run on the PC, Portable PC, PC/XT, and PC/AT with IBM's new 3½-inch external diskette drive option. The system can process as many as 128 diskettes per hour in the PC's 5¼-inch format and 69 diskettes per hour in the Convertible's 3½-inch format. $29,350.
Media Systems Technology, Inc., 16812 Hale Avenue, Irvine, CA 92714; 714/863-1201
CIRCLE 307 ON READER SERVICE CARD

The **LaserPro EXPRESS** is an eight-page-per-minute laser printer that emulates the Hewlett-Packard LaserJet, the Diablo 630, the Qume Sprint 11, and the Epson FX-80. Introduced by **OASYS** (Office Automation Systems, Inc.), the LaserPro EXPRESS features a 250-sheet paper cassette for 8½-by-11-inch paper, a 100-sheet output capacity tray, and a 50-sheet adjustable paper stacker that accepts paper as small as 4¼ by 5½ inches and as large as 8½ by 14 inches. The user can control whether printed output emerges with the printed side facing up or down. The printer is equipped with a Mita engine and provides 384KB of standard memory. It also features 10 bit-mapped fonts that yield 72 font variations as a result of bolding, italicizing, and compression. $1,895.
OASYS, 8352 Clairemont Mesa Blvd., San Diego, CA 92111; 619/576-9500
CIRCLE 308 ON READER SERVICE CARD

Alloy Computer Products, Inc. has introduced **Bi-TURBO**, an accelerator board that provides dual-tasking software and additional memory to allow users to perform two tasks simulta-

This looks like our solution

iti®

INVEST IN ASSETS TO BUILD ON

LinkUp Products give you . . .

- System Approach
- Proven Reliability
- Ease of Installation
- An Easy Transition from Existing Systems
- A Minimal Training Curve
- User-friendly Software
- Upgradability
- Toll-free Technical Assistance

iti®
INFORMATION TECHNOLOGIES INC.

Corporate Headquarters
7850 East Evans Road • Scottsdale, Arizona 85260
602/998-1033 • 800/431-3460

5-1, increases
e computer by
d with the
cache, print
ASTROM to
d IBM BASICA
ures a battery-
two I/O ports.
ible in two

Alloy Computer Products, Inc., 100 Pennsylvania Avenue, Framingham, MA 01701; 617/875-6100

CIRCLE 315 ON READER SERVICE CARD

The **10-TEST Diagnostic Tool**, from **Fox Research, Inc.**, is designed to aid 10-NET users in tracking network hardware problems. It provides easy testing of network boards and general network integrity. The package consists of a diagnostic module that can be connected directly to a board in a PC or to any tap box in the network, a transformer to provide power to the module, and diagnostic software on diskette. During testing of a 10-NET board in a PC, any detected error and possible causes of failure are displayed on the screen. $695.
Fox Research, Inc., 7016 Corporate Way, Dayton, OH 45459-4223; 513/433-2238

CIRCLE 312 ON READER SERVICE CARD

Model NMS 8003B is a 368MB Winchester disk system that is interfaced to the CPU using the NMS high-speed single board disk/tape controller. From **National Memory Systems Corporation**, the 8003B comes with a volume formatter program that allows the memory system to be used as one logical volume under DOS. The 8003B boasts disk data rates of nearly 2MB per second and average access times of 15 milliseconds. The system is compatible with most networks and supports XENIX. $9,000.
National Memory Systems Corporation, 355 Earhart Way, Livermore, CA 94550; 415/443-1669

CIRCLE 319 ON READER SERVICE CARD

drives, providing 90MB of fixed and removable on-line storage. The chassis fits between the monitor and the CPU of the XT and AT. From $2,499 to $3,695.
SyQuest Technology, 47923 Warm Springs Blvd., Fremont, CA 94539; 415/490-7511

CIRCLE 314 ON READER SERVICE CARD

A LAN board designed to use the 16-bit bus of the PC/AT and RT/PC has been announced by **The Destek Group**. The **NIB100/08** features 16-bit data transfers from the host and incorporates 48KB of on-board buffer memory. The board uses the Destek 2 megabits-per-second CSMA technology and incorporates an on-board CPU with its own memory for network management. An on-board optically isolated transceiver gives the network a 3,000-foot distance without repeaters, in addition to greater electrical immunity. The desNET BIOS program makes it 100-percent software compatible with NET BIOS. $650; evaluation kit sufficient for two nodes, $950.
The Destek Group, 630 E. Evelyn Avenue, Sunnyvale, CA 94086; 408/737-7211

CIRCLE 305 ON READER SERVICE CARD

Univation, Inc. has introduced the **Dream Board**, a single add-in circuit board that combines the functions of a multifunction board, an EMS board, and an accelerator board. The board's memory module can be expanded to a maximum of 2MB of RAM that is fully compatible with the Lotus/Intel/Microsoft expanded memory specification (LIM EMS). The accelerator module, incorpo-

configurations: the Turbo EMS, $795, and the Turbo multifunction, $995.
Univation, Inc., 1231 California Circle, Milpitas, CA 95035; 408/263-1200

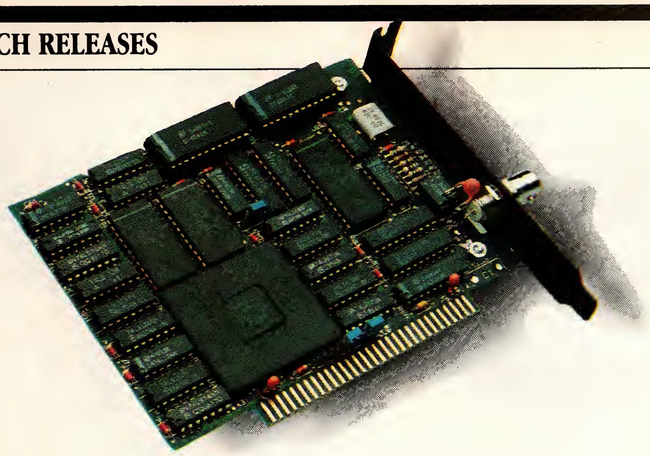
CIRCLE 313 ON READER SERVICE CARD

A memory expansion board for the RT/PC has been announced by **Tall Tree Systems**. The **JRAM-RT** uses 1MB chips to provide 8MB on a single board, and it uses only one expansion slot. This 32-bit board has error correcting code that uses the motherboard facilities. No software or hardware modifications or switch setting is necessary. $3,995.
Tall Tree Systems, 1120 San Antonio Road, Palo Alto, CA 94303; 415/964-1980

CIRCLE 304 ON READER SERVICE CARD

Two CD-ROM drives have been announced by **Sony Corporation of America**. The **CDU-5002** is designed to fit into 5¼-inch floppy disk slots, and the **CD-100** is a stand-alone drive with its own power supply. The CD-100 is equipped with two 40-pin Sony bus connectors and can be organized into a daisy chain of as many as four drives. Both Sony drives come with system controllers and perform descramble, synchronization detection, header check, and error correction functions. The drives also have buffer memories equivalent to one CD block (2KB). CDU-5002, $965; CD-100, $1,065.
Sony Corporation of America, Corporate Communications Department, Sony Drive, Park Ridge, NJ 07656; 201/930-1000

CIRCLE 316 ON READER SERVICE CARD



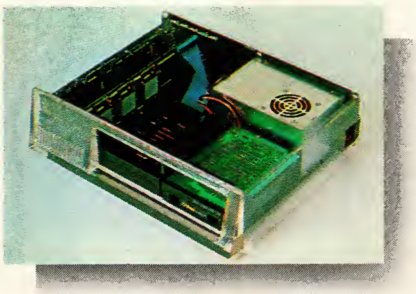
Quadram's MainLink Plus coaxial board

A DRI Concurrent PC-DOS Expanded Memory screen

Quadram Corporation has launched its **MainLink** line of micro-to-mainframe communication products by announcing four new 3278/79 emulation products. The new MainLinks are half-card expansion boards that permit a PC, PC/XT, or PC/AT to emulate 3278/79 terminals. All four products support a uniform application program interface. The MainLink Standard coaxial connection and the MainLink Plus coaxial connection link directly to an IBM 3274 or 3276 cluster controller for local or remote processing. Both are IRMA compatible. The MainLink Standard remote and the MainLink Plus remote attach via synchronous modem to an IBM 3705, 3725, or equivalent communications controller in SNA/SDLC mode. Both permit emulation of an IBM 3274 cluster controller and 3287 host-addressable printer. The MainLink Plus products feature a windowed control program, permitting simultaneous access to one host session, a DOS session, and two electronic notepads. They also include a high-performance file-transfer capability for TSO or CMS host systems. MainLink Standard coaxial connection, $895; Plus coaxial, $1,145. MainLink Standard remote, $545; Plus remote, $985.
The Quadram Communication Products Group, Asher Technologies, Inc., 1009 Mansell Road, Roswell, GA 30076; 404/993-4590
CIRCLE 318 ON READER SERVICE CARD

Targa Electronics Systems, Inc. has introduced the **Solidrive Diskless Storage Architecture**, a family of products designed for the IBM PC and industrial PCs. The family includes a full-sized board with an expansion bus connector and .5MB, 1MB, or 1.5MB of bubble memory; expansion packs with 1MB or 2MB of bubble memory; and Targa's Solidrive bubble memory cartridge systems with up to .5MB of removable bubble memory. Expansion units can be daisy-chained to provide 8MB of bubble memory as an extension to the 1.5MB on the main board. Thus, the new architecture can provide up to 9.5MB of bubble storage capacity.

The products use at most one expansion slot in the host computer and allow for a combination of fixed and removable storage. They incorporate the new Intel 4Mbit bubble memory devices. The board appears to the operating system as a Winchester disk and can be configured to be one of two possible hard drives. Features of the bubble memory board include an RS-232 communications port, a password-protection scheme, switchable write protection, built-in test, diagnostic LEDs, configurable EPROM base address, interrupt,



Targa's Solidrive

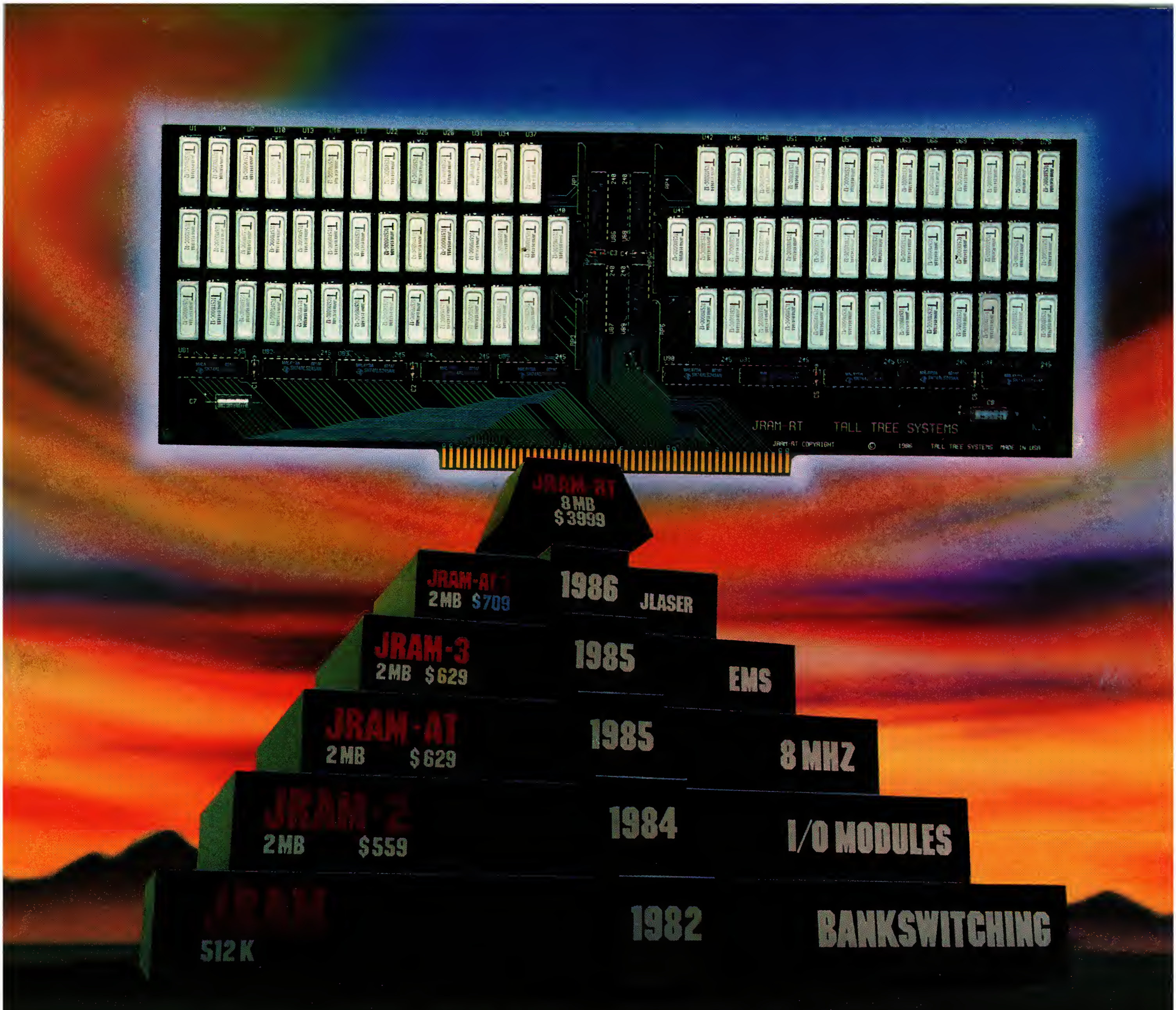
and DMA channels. The board is available in normal (10 to 45 degrees centigrade) and extended (0 to 65 degrees centigrade) temperature range versions. Normal range board with .5MB of memory, $1,195; with 1MB, $1,995; with 1.5MB, $2,955; expansion packs with 1MB of memory, $1,995; with 2MB, $3,915; removable bubble cartridges with .5MB capacity, $1,275; cartridge holder, $315. Extended range board with .5MB of memory, $1,595; with 1.5MB, $3,695. (All for a quantity of 50.)
Targa Electronic Systems, Inc., 3101 Hawthorne Road, P.O. Box 8485, Ottawa, Ontario, Canada K1G 3H9; 800/267-9793; in Canada, 613/731-9941
CIRCLE 309 ON READER SERVICE CARD

SOFTWARE

Concurrent PC-DOS Expanded Memory (XM), which incorporates expanded memory and multitasking and runs existing PC-DOS, MS-DOS, Concurrent DOS, and CP/M 86 applications without modification, has been introduced by **Digital Research, Inc.** The new operating system supports the LIM and AQA expanded memory specifications. With the expanded memory, Concurrent PC-DOS XM can accomodate three users, providing a low-cost alternative for multiuser computing that can be networked. A configurable memory system, menu-driven file manager, an on-line help facility, and a full set of accessory products are included in the software package. $395.
Digital Research, Inc., Box DRI, Monterey, CA 93942; 408/649-3896
CIRCLE 320 ON READER SERVICE CARD

Norton-Lambert Corporation has introduced **Close-Up**, a remote communications system that allows two computers to operate as one. At the same time, the host computer is free to operate normally. Close-Up features remote printing, fast file transfer, realtime color graphics support, total error protection, movable chat windows, built-in time and billing systems, and the ability to disable a customer keyboard. Support side, $245; customer side, $195.
Norton-Lambert Corporation, P.O. Box 4085, Santa Barbara, CA 93140; 805/687-8896
CIRCLE 322 ON READER SERVICE CARD

Mbp Software and Systems Technology, Inc. has announced **Visual COBOL**, an enhanced version of its native code COBOL compiler. The new compiler features an improved integrated screen management system and a compilation speed that is 50 percent faster than its predecessor. Also included



TALL TREE SYSTEMS. A Technological Innovator. Always a Step Ahead!

For true industry leadership, look no further than Tall Tree Systems.

We have a history of being first.

We were the first to introduce bankswitching. The first with two megabyte memory boards. The first with I/O modularity in a single slot. The first with 8 MHz speed capabilities. The only maker of single command EMS boards. The first with a laser printer solution — JLASER — that allows you to do full-page graphics and multiple type fonts on any Canon® or Ricoh® laser engine.

Now, we're first again with memory expansion for the IBM®RT.

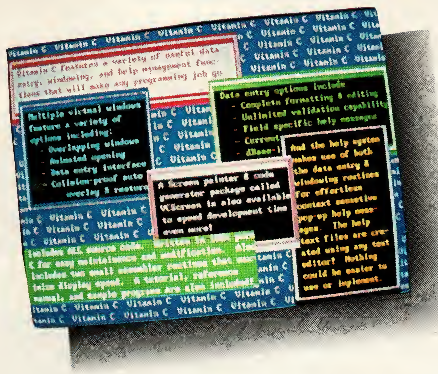
Innovation is our tradition.

Our trademark is superior technology at the lowest possible price.

TALL TREE SYSTEMS

1120 San Antonio Road • Palo Alto, CA 94303 • (415) 964-1980

CIRCLE NO. 197 ON READER SERVICE CARD



A screen from CPC's Vitamin C

SyncTalk communications software from NSA

are a fast SORTs, a shrink utility that can reduce the size of executable code by more than 50 percent, full support of DOS path names, and revised documentation. The compiler size also has been reduced 50 percent. $1,150.
mbp Software and Systems Technology, Inc., 1131 Harbor Bay Parkway, Suite 260, Alameda, CA 94501; 415/769-5333
CIRCLE 328 ON READER SERVICE CARD

Finally!, a library of more than 100 named subroutines for use in compiled BASIC programs has been introduced

KOMPUTERWERK's subroutine library for compiled BASIC

by **KOMPUTERWERK**. It takes advantage of the named subroutine capability in the newer compilers, such as IBM BASIC 2.0 or Microsoft QuickBASIC. Subroutines in Finally! can be used for averaging and determining maximum and minimum values in arrays; loading disk directories into an array; changing the default drive; stripping leading and trailing blanks from strings; sorting numeric and string arrays; drawing boxes, pie, and bar charts; and determining memory and peripheral configuration. Source code for all subroutines and functions is included and all are fully documented and indexed. Finally! also includes a cross-reference utility. $99.
KOMPUTERWERK, 851 Parkview Blvd., Pittsburgh, PA 15215; 412/782-0384
CIRCLE 325 ON READER SERVICE CARD

Network Software Associates, Inc. (NSA) has announced a new peer-to-peer synchronous communications and file-transfer program for the PC and compatibles. **SyncTalk** lets two PCs communicate and exchange files at speeds of as high as 9600 bps, using IBM's SDLC (Synchronous Data Link Control) protocol. SyncTalk enables either PC to initiate and control the exchange on a dynamic basis. Both PCs run an identical version of the software. SyncTalk also supports selected remote DOS commands issued locally and executed automatically on the remote PC, thereby providing for remote control of an unattended PC. $195.
Network Software Associates, Inc., 22982 Mill Creek, Laguna Hills, CA 92653; 714/768-4013
CIRCLE 321 ON READER SERVICE CARD

CompuFirm Corporation has introduced **PL/M CONNECTION**, a comprehensive interface library of more than 150 functions and utilities, providing a direct connection between the PC and Intel's PL/M compiler. PL/M CONNECTION enables a developer to access all of the PC's DOS and BIOS functions, plus high-speed graphics functions that interface directly with the PC's color graphics and monochrome display adapters. The package also includes development support for software control of PC peripherals such as floppy and hard disks, communications devices, printers, video, and keyboard. $295.
CompuFirm Corporation, 7677 Ronson Road, Suite 204, San Diego, CA 92111; 619/571-0228
CIRCLE 337 ON READER SERVICE CARD

The first COBOL for the RT/PC has been announced by **Micro Focus**. The product set includes **Compact Level II COBOL/ET**, a high-quality ANSI 74 COBOL compiler, **ANIMATOR**, **FORMS-2**, and **Upgrade III**. The compiler is certified at high level with zero errors by the GSA. With a time run library, it produces code that can be used in the RT multiuser environment. The compiler generates intermediate code from source programs created using a standard UNIX editor, such as **vi**. Support is provided for very large programs, all four COBOL file formats, and calls to programs written in C, the RT system programming language. ANIMATOR migrates applications from one environment to another. FORMS-2 is a screen painter and prototyping tool. Compact Level II COBOL/ET, $2,000; ANIMATOR, $1,200; FORMS-2 and Upgrade III both for $400. The COBOL Developer's Toolkit (all products together), $3,750.
Micro Focus, 2465 E. Bayshore Road, Palo Alto, CA 94303; 415/856-4161
CIRCLE 327 ON READER SERVICE CARD

Creative Programming Consultants, Inc. (CPC) has released **version 2.05** of the **Vitamin C** function library, which supports the new Lattice 3 and Wizard C compilers. Vitamin C features comprehensive data entry, windowing, and application help management routines. Data entry capabilities include right-to-left numeric input, picture clause formatting, and complete validation. Windowing functions offer automatic overlay and restore with support for multiple virtual windows. The help management system is integrated with the data entry and windowing routines. $149.95; updates for 2.0, $10.00; for earlier versions, $50.00.

The company also produces **VCScreen**, which is an interactive screen painter and code generator that is fully compatible with Vitamin C. Using VCSreen's interactive editor, the programmer can see exactly how finished screens will look throughout the entire design process. $99.95.
Creative Programming Consultants, Inc., P.O. Box 112097, Carrollton, TX 75011-2097; 214/245-6090
CIRCLE 333 ON READER SERVICE CARD

Mark Williams

Now the biggest name in C compilers comes in a size everybody can afford.

Let's C.™

$75

Introducing Mark Williams' $75 C compiler. Want to explore C programming for the first time? Or just on your own time? Now you can do it in a big way without spending that way. With Let's C.

This is no little beginner's model. Let's C is a powerful programming tool, packed with all the essentials of the famous Mark Williams C Programming System. The one chosen by Intel, DEC, Wang and thousands of professional programmers. The one that wins the benchmarks and the reviewers' praise:

"(This compiler) has the most professional feel of any package we tested..."—BYTE

"Of all the compilers reviewed, (it) would be my first choice for product development."—David W. Smith, PC WORLD

And now for more big news. Get our revolutionary csd C Source Debugger for just $75, too. You can breeze through debugging at the C source level ignoring clunky assembler code.

Affordable, powerful, debuggable. Mark Williams Let's C is the big name C compiler at a price you can handle. Get your hands on it now.

Mark Williams Let's C

- **For the IBM-PC and MS-DOS**
- **Fast compact code plus register variables**
- **Full Kernighan & Ritchie C and extensions**
- **Full UNIX™ compatibility and complete libraries**
- **Small memory model**
- **Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc**
- **MicroEMACS full screen editor with source**
- **Supported by dozens of third party libraries**
- **Upgradeable to C Programming System for large scale applications development**

Let's C Benchmark Done on an IBM-PC/XT, no 8087. Program: Floating Point from BYTE, August, 1983.

	Exec Time in Seconds
Let's C	134.20
MS 3.0	347.45

Use this coupon or charge by calling toll-free: 1-800-MWC-1700. In Ill. call 312-472-6659.

ORDER NOW! 60-DAY MONEY BACK GUARANTEE!

Mark Williams Let's C

Please send me:

_____ copies of Let's C and _____ copies of csd (C Source Debugger) at $75 each. (Ill. residents add 7% sales tax.)

☐ Check ☐ Money Order ☐ Visa, MasterCard or American Express

Name_____

Address_____

City_____ State_____ Zip_____

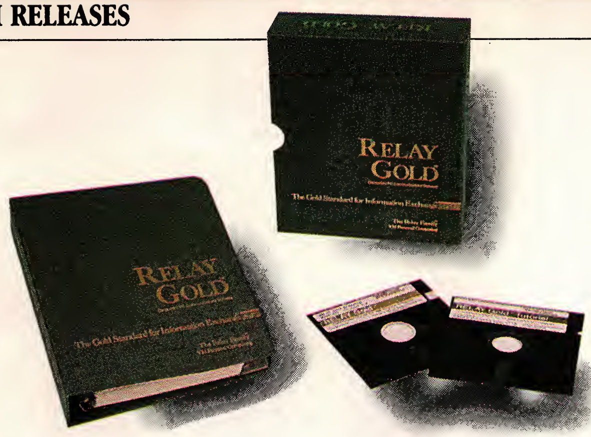
Card #_____ Exp. Date_____

Signature_____

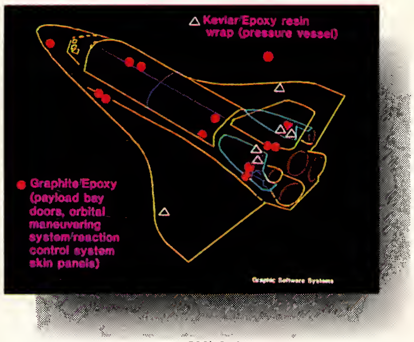
TJ0986

Mark Williams Company
1430 West Wrightwood
Chicago, Illinois 60614

CIRCLE NO. 205 ON READER SERVICE CARD



Version 2.0 from VM Personal Computing



GSS*GKS screen

VM Personal Computing has announced **version 2.0** of **RELAY Gold** and a new communications product, **RELAY Silver**. RELAY Gold 2.0 now supports IRMA, IBM, and Forte 3270 emulation boards and most protocol converters and allows error-free, high-performance file transfer from mainframes to PCs running RELAY Gold with these emulation boards and VMPC's mainframe software. The upgraded RELAY Gold is now memory resident, and the addition of API (Application Program Interface) allows programmers to run RELAY Gold from within their own programs. RELAY Gold, $250; upgrade for 1.0 purchased prior to June 9, 1986, $40.

RELAY Silver is an advanced PC software package for PC-to-PC and PC-to-information service communications. It has all of RELAY Gold's PC communications features, including the learn mode, script language, background operation, and the API; however, it is not capable of communicating with VMPC's mainframe products. $150.
VM Personal Computing, 41 Kenosia Avenue, Danbury, CT 06810; 800/222-VMPC; in Connecticut, 203/798-3800

CIRCLE 331 ON READER SERVICE CARD

The **GSS*GKS Graphical Kernel System** from **Graphic Software Systems, Inc.** (GSS) enhances RT/PC graphics programming and enables the porting of mainframe, minicomputer, and microcomputer graphics applications to the RT. GSS and IBM previously announced IBM's licensing of the VDI-based RT/PC Graphics Development Toolkit (GDT) and several other RT packages. GSS*GKS adds a higher-level development environment to IBM's UNIX-based RT/PC GDT. The packages are source-code compatible with the DOS-based PC, PC/XT, PC/AT, and 3270-PC versions IBM has licensed from GSS during the past two years. GSS*GKS supports the IBM Enhanced Graphics Adapter, the Extended Monochrome Graphics Adapter, many IBM plotters and printers, and the RT Mouse. $795.
Graphic Software Systems, Inc., 9590 S.W. Gemini Drive, P.O. Box 4900, Beaverton, OR 97005; 503/641-2200

CIRCLE 334 ON READER SERVICE CARD

A DOS shell, **The Norton Commander** from **Peter Norton Computing, Inc.**, offers complete access to all DOS functions, including full use of the command line and screen. The Commander supports single and multiple file operations and features integral editing and viewing. The shell implements single-key execution of program and data files, user-defined menus of commands, and full support of all Microsoft-compatible mouse devices. $75.

A new DOS shell from Peter Norton Computing, Inc.

Peter Norton Computing, Inc., 2210 Wilshire Blvd., Suite 186, Santa Monica, CA 90403-5784; 213/453-2361

CIRCLE 323 ON READER SERVICE CARD

A multipurpose tool from **Pacific Crest Software**, **POINT FIVE** can be used for both numerical and graphical modeling and analysis across a wide range of disciplines and functions. At its most fundamental level, POINT FIVE operates as a programmable calculator with 150 built-in mathematical, financial, statistical, and data manipulation functions. By using these functions as a fourth-generation language, users can create sophisticated programs. $195.
Pacific Crest Software, 887 NW Grant Avenue, P.O. Box 2098, Corvallis, OR 97339; 503/754-1067

CIRCLE 335 ON READER SERVICE CARD

OTHERWARE

A CD-ROM evaluation kit is being offered by **Computer Access Corporation** and **Videotools**. With **BlueFish** software on a PC, any user-supplied ASCII text becomes an electronic book in which every word is indexed and the information can be rapidly sorted or searched on a CD-ROM. BlueFish records are completely interactive. Searching a BlueFish CD-ROM locates all relevant documents and delivers them in electronically reusable form. Processed BlueFish files are formatted on a VideoTools CD Publisher system that produces the one-half-inch data tape, the medium for premastering.

After a quality control check, the tape produced by the VideoTools system is then shipped to Eindhoven in the Netherlands where the CD-ROM is premastered, mastered, and replicated. The kit includes BlueFish software and user license, VideoTools preparation service, with premastering, mastering, and replication of 50 custom CD-ROM copies of up to 50MB of the user's own data by Philips Subsystems and Peripherals, Inc. The purchase price also includes a professional-quality Philips CM 100 CD data player, card, and driver software for operation with the user's PC. $9,800.
Computer Access Corporation, 26 Brighton Street, Suite 324, Belmont, MA 02178-4008; 617/484-2412

CIRCLE 338 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

"One of the primary reasons the Computer Press Association was formed was to promote high standards of writing in computer journalism. It's time to reward those who do it best."

Hal Glatzer
Vice President
Computer Press Association


Computer Press
Awards
1985
Best Computer Magazine
PC Tech Journal
CITIZEN
Computer Press
Association

Best Computer Magazine

We are proud and honored to have been selected as the "Best Comput Magazine" by the Computer Press Association in its first annual awards.

At the awards ceremony, it was said that "PC TECH JOURNAL has substance, style and clear writing combined with superior information, value and contemporary graphics."

Indeed, it would be difficult for us to express more succinctly the standards we set for ourselves in publishing PC TECH JOURNAL. And we accept with appreciation the fact that our professional colleagues have recognized our achievement.

By concentrating on the rapidly growing need for information about PC systems, PC TECH JOURNAL has created a unique publication for PC systems experts, the key segment of an extremely sophisticated market.

This award as the "Best Computer Magazine" commits us even more toward the development of PC TECH JOURNAL as the information source for those dedicated systems professionals whose vision and industry are fast making the personal computer the most important business tool ever invented.

THE PROGRAMMER'S SHOP™

31 Day
RISK-FREE TRIAL
on any product in this ad.

C Programmers:
7 Ways to Increase Productivity

Add Efficient MultiTasking to Your C Programs with
Multi-C

Multi-C lets you create, manage, and communicate among tasks with little RAM and little processor overhead. Now you can handle multiple users, printers, communications, or just about anything else in a logical, well structured manner, without complex polling schemes or lockups in your programs.

Unlike an operating system toolkit, Multi-C is designed for ease of use in production programming, and Multi-C **cooperates** with your operating system, so functions are reliable. Existing and new programs can add Multi-C features without paying royalties. 40+ functions use 32,000 priority levels, handle interrupts, control flags, and messages and control queues.

Multi-C is lower in cost than other products of this type, and it's portable because it's written in C; use it for stand-alone systems. You can even make Multi-C the kernel of your own operating system!

Particularly good for applications in data acquisition, device control, communications, and to give the user control of application use, response, and keyboard productivity. Multi-C has all the functions you need for task scheduling, time scheduling, and inter-task communication.

Specify for use with MS C, Lattice 2.15 or 3.0, or C-86. Partial source.

Cytek **MSDOS $149**

First Database Exclusively for C is also Royalty-Free.
db_VISTA from RAIMA Corporation

— *"If you are looking for a sophisticated C programmer's database, this is it."* —

- Dave Schmitt, President, Lattice, Inc.

Designed exclusively for C, db_VISTA is a royalty-free programmer's DBMS. Take full advantage of C through ease of use, portability, and efficiency. You optimize for speed and efficient disk storage.

Multiple key records, fast B-tree indexing, virtual memory disk accessing. Tailor db_VISTA to your needs by using only those features you require. ASCII file transfer utilities make moving to db_VISTA a snap.

MSDOS, Unix, Xenix, Macintosh, Amiga. Single user Source $459 Object $179. Multiuser Source $929. Object $450

Even for Small Files: Convenient, Fast Access
CBTREE — Only $99

Why spend time writing file management code when you can use consistent, flexible, documented, professional functions? Even multiuser record locking and variable-length records are supported.

Add, delete, and update without needing to reindex. Store keys and record locations in B+trees.

You can access any record or group of records by the value of a user specific key. Search your files from any point, forward or backward.

Full, balanced b-tree support includes use of multiple keys, unlimited number and length of keys.

Use this powerful ISAM, even if you've previously done without.

Learn how to write systems for managing large files by using CBTREE source as a guide. Modify it and transfer it to another operating environment without royalties.

MSDOS $99

Full Screen Library Plus Fundamentals
THE HAMMER LIBRARY

1-2-3 menus (even multilevel), data validation including range checking, field editing, and Full Screen input make this a thorough access library. Data entry routines let you move between fields before completing screen.

But you also get DOS and BIOS access, file handling, string functions, and data handling. Plus pattern matching calls and library handling utilities.

Full source, no royalties. Data entry routines for solid interface with user. Specify MS C, Lattice, CI-C86, Desmet, MW C.

MSDOS. A lot for only $179

Looking for the right tool for the job? Reach for THE HAMMER.

Flexible Screen Development with SECURITY CHECKING and HELP SCREENS:
ZVIEW Screen Library

Use this field-sensitive tool to develop data entry screens and windows and provide run-time flexibility. Security level settings restrict inquiry or update of fields; multiple screen help display is available at screen and field level. You can also customize ZVIEW's operation and make any field characteristic change during execution.

ZVIEW gives you full control of attributes, colors, boxes, protected fields, scrolling, and more. Load screens from memory for fast response. Field support includes alpha, numeric, or alphanumeric data types, case conversion, range checking, and field comparison, and ZVIEW provides automatic data conversion to and from ASCII screen format. For Microsoft C, Lattice 3.0, and Aztec 3.2e. Supports EGA, color, and monochrome displays.

PCDOS $219

C DYNAMO! WINDOWING: Full C Source, No Royalties
POWER WINDOWS AND C FUNCTION LIBRARY

Power Windows covers all the bases; overlays, borders, 1-2-3 style or pop-up menus/help windows, zap instantly on/off screen, status lines, horizontal/vertical scrolling, color control or highlighting, word-wrap, files to windows, keyboard to windows. Powerful, easy to use, integrated error messages, thorough documentation. Supports IBM monochrome or color.

MSDOS. Only $119

C Function Library - includes 325 fundamental functions with readable source and thorough documentation.

MSDOS. Only $119

No matter what you have, you need these. Best value available. Highly recommended.

Fastest C Development on Earth:
Instant C Version 2.0

Instant C's NEW version 2.0 gives you immediate (2 secs.) compilation and execution of large (5,000+ line) programs, and the ability to link in external (commercial or your own) libraries in an interactive, Lattice 3.0 or Microsoft 3.0 compatible, interpreter-like environment with an integrated full screen editor and source level debugger.

You get full K&R standard C, fast (33 second sieve) execution speed, and debugging with source code animation, single-stepping, backtracing, and unlimited conditional breakpoints.

Instant C now supports multiple screens and graphic devices, run-time checking of pointer and array references, and includes a new manual, expanded tutorial and reference section, and complete library source.

MSDOS $399

Rational Systems, Inc.

Call for a catalog, literature, advice and service you can trust

NEW HOURS
8:30 AM - 8:00 PM EST.

800-421-8006

THE PROGRAMMER'S SHOP™
128-P Rockland Street, Hanover, MA 02339
Mass: 800-442-8070 or 617-826-7531 5/86

"YOUR SERVICE IS FABULOUS!!! . . . Your information packets are so enticing; I really want 10 of them . . . Keep up the good work."
— Dana Bredemeyer
Bredemeyer Company

CIRCLE NO. 220 ON READER SERVICE CARD

MICHAEL ABRASH

Flicker-free Scrolling

BIOS-related flicker can be eliminated on the IBM Color Graphics Adapter and its clones.

When scrolling in text mode, the IBM Color Graphics Adapter (CGA) flickers annoyingly. Because the CGA's display memory is not dual-ported, snow appears on the screen during block memory moves. To hide this snow, BIOS disables video while scrolling and produces the flickering. Many CGA clones *are* dual-ported and do not generate snow; unfortunately, the BIOS always disables video during scrolling, causing these cards to flicker just like the CGA.

SCROLLER.ASM intercepts the **write_tty** BIOS function when scrolling in mode 2 or 3 and scrolls with a routine that does not disable video. The remaining flicker results from scrolling with the **scroll_up** and **scroll_down** functions; if fully flicker-free scrolling is needed, these functions can be replaced with an expanded SCROLLER.ASM.

Michael Abrash is a senior software engineer for Orion Instruments.

LISTING 1: SCROLLER.ASM

```
; Resident program to provide flicker-free write_tty scroll for Color
;  Graphics Adapter clones with dual-ported memory.  M. Abrash 5/3/86.
; Make runnable with MASM-LINK-EXE2BIN.
cseg    segment
        assume  cs:cseg
        org     100h    ;necessary for COM file
start   proc    near
        jmp     makeres
old_int10       dd      ?
; front end routine for BIOS video handler to scroll without flicker
scroll_front_end:
        cmp     ax,0e0ah        ;only intercept write_tty function
        jnz     pass_to_bios    ; called with linefeed
        push    ax
        push    bx
        mov     ah,0fh
        int     10h             ;get current page & mode
        cmp     al,2
        jz      check_row       ;BIOS only blanks in modes 2 & 3, so
        cmp     al,3            ; only intercept linefeed scroll in
        jnz     pass_to_bios2   ; modes 2 & 3
check_row:                      ;see if cursor is on bottom row, in
        push    cx              ; which case linefeed causes scroll
        push    dx
        mov     ah,3
        int     10h             ;get cursor location in current page
        cmp     dh,24
        jnz     pass_to_bios3   ;cursor not on bottom row, no scroll
        push    ds              ;meets all the criteria, so perform
        push    es              ; scroll in current page with special
        push    si              ; routine that doesn't disable video
        push    di
        mov     ah,0fh
        int     10h             ;get # columns & page
        mov     al,ah
        sub     ah,ah   ;convert to word
        push    ax      ;set aside # columns
        mov     si,ax
        shl     si,1    ;move from second row (each character=2 bytes)
        mov     ah,24
        mul     ah      ;# words to move (24 rows)
        mov     cx,ax
        sub     ax,ax   ;now adjust offsets for current page
        mov     ds,ax   ;buffer length is stored in BIOS segment
        mov     al,bh   ;get current page
        mul     word ptr ds:[44ch] ;offset of start of current page
        add     si,ax   ;move data from second row of current page
        mov     di,ax   ; to top of current page
        mov     ax,0b800h
        mov     ds,ax
        mov     es,ax           ;will move data in display segment
        cld
    rep movsw           ;scroll screen up
        mov     ah,8    ;BH already has current page
        int     10h     ;get attribute of character at cursor
        mov     al,' '  ;fill with blanks & attribute just obtained
        pop     cx      ;# of words per row
    rep stosw           ;blank bottom row-DI points to bottom row
        pop     di      ;done
        pop     si
        pop     es
        pop     ds
        pop     dx
        pop     cx
        pop     bx
        pop     ax
        iret
pass_to_bios3:
        pop     dx
        pop     cx
pass_to_bios2:
        pop     bx
        pop     ax
pass_to_bios:           ;pass interrupt to normal BIOS handler
        jmp     cs:[old_int10]
endres:
; make scroll front end handler resident & revector interrupt 10 to it
makeres:
        push    cs
        pop     ds
        assume  ds:cseg
        mov     ax,3510h        ;DOS get vector function, vector 10h
        int     21h             ;get vector 10h
        mov     word ptr [old_int10],bx ;set aside old vector to
        mov     word ptr [old_int10+2],es       ; allow pass to BIOS
        mov     ax,2510h        ;DOS set vector function, vector 10h
        mov     dx,offset scroll_front_end      ;revector interrupt
        int     21h                             ; 10h to front end routine
        mov     dx,offset endres ;# of paragraphs to make
        mov     cl,4            ; resident-can't do with an
        shr     dx,cl           ; expression because assembler can't
        inc     dx              ; calculate w/relocatable label
        mov     ax,3100h        ;DOS make resident fn, exit code=0
        int     21h             ;terminate & stay resident
start   endp
cseg    ends
        end     start
```

ONE STOP.

PROGRAMMER'S

QUALITY INTEGRITY

Service

SUPPORT PRICE

CONNECTION

We're Programmer's Connection, a leading independent dealer of quality programmer's development tools specifically for IBM Personal Computers. We're your best one-stop source for the professional PC/MS-DOS and XENIX programming tools you need.

Since we're an *independent* dealer, we'll look out for your best interest. Our courteous, knowledgeable, noncommissioned sales staff is always ready to assist you. If you aren't sure about your needs, you can talk to our experienced, professional technical staff for sound, unbiased advice. They can compare products, answer technical questions and send you detailed product information that's tailored to your needs.

Our product line consists of hundreds of high quality software development tools specifically for IBM Personal Computers and compatibles. We don't carry *everything* ever written for programmers — only those products that meet our very high standards for quality and value.

The products we carry are the latest version and they all come with the same manufacturer's technical support as if buying direct. Unlike other dealers who participate in the software gray market, we're authorized to sell *every* product we carry.

We discount all software products — even special order items. We don't try to fool you by discounting some products and charging full list price for others. Every product in our price list is shown with its discounted *and* retail price. We want you to know exactly how much you'll save on *every* product.

Other dealers add extra rush charges for shipping via express services. We'll express your order to you with no special handling charges. We only charge you what the shipping carrier charges. And if you have your order shipped via standard UPS, shipping is *FREE*.

Most popular products are in stock and ready for immediate shipment. Maintaining an adequate inventory is part of our philosophy of fast, efficient service. If we don't have a product in stock, we'll get it for you fast. Again — no extra charge.

Some dealers charge your credit card at the time they take your order. This means you could be left waiting in vain for your order for weeks or months while they use your money interest free. We *never* charge your credit card until we actually ship your order.

Quite simply, the discount prices you see on the next two pages are all you pay. There are *no* hidden charges. We don't charge extra for standard UPS shipping, credit cards, COD orders, purchase orders or special handling (except orders outside the U.S. and Canada are charged $5 for customs form preparation).

Our goal is customer satisfaction and that's why we offer 30 day no-risk return guarantees and 30 day evaluation periods on most of our products. Note that some items, especially those with source code, are restricted by the manufacturers from this guarantee. Please call for specific details.

Our customers keep coming back because we consistently provide high quality service and low discount prices. So make the connection today and see how convenient one-stop service can be.

CALL TOLL FREE

US: 800-336-1166 ▪ CANADA: 800-225-1166

OHIO AND OVERSEAS: 216-877-3781

Hours: 8:30 AM to 8:00 PM Eastern Time, Mon.-Fri.

Customer Service: 216-877-1110

PC programmer's connection

Call or write for our *FREE* comprehensive price guide. Turn the page for our latest advertised price list.

PROGRAMMER DEVELOPMENT TOOLS FOR THE IBM-PC/XT/AT
and compatibles.

apl language

	LIST	OURS
APL*PLUS/PC System by STSC	595	449
APL*PLUS/PC Tools Vol 1 by STSC	295	239
APL*PLUS/PC Tools Vol 2 by STSC	150	129
APL*PLUS Spreadsheet Manager by STSC	195	159
APL*PLUS/UNX System *For AT Xenix* by STSC	995	795
Btrieve *ISAM File Mgr with No Royalties* by SoftCraft	250	195
Financial/Statistical Library by STSC	275	219
Pocket APL by STSC	95	79
STATGRAPHICS *Statistical Graphics System* by STSC	795	619

arity products

Arity Expert System Development Package *New*	295	279
Arity File Interchange Toolkit *New*	50	48
Arity Prolog Compiler & Interpreter *New*	795	739
Arity Prolog Interpreter *New*	350	329
Arity SQL Development Package *New*	295	279
Arity Screen Design Toolkit *New*	50	48
Arity Standard Prolog *New*	95	89

artificial intelligence

ExpertEDGE by Human Edge	795	659
Experteach II *Complete System* by Intelligenceware	475	CALL
EXSYS *Expert System Development Software* by EXSYS	395	339
First Class by Human Edge *New*	500	419
GCLISP *Golden Common LISP* by Gold Hill	495	CALL
GCLISP 286 Developer *LM Interpreter & LM Compiler*	1190	CALL
Insight 1 *AI Primer* by Level Five Research	95	75
Insight 2+ by Level Five Research	485	389
Logic-Line Series *All varieties* by Thunderstone *New*	CALL	CALL
Microsoft LISP *Common LISP*	250	189
PLA microProlog by Programming Logic Associates	250	219
with APES	450	399
PLA Professional microProlog by Programming Logic	395	349
with APES	695	599
QNIAL *Combines APL with LISP* by NIAL Systems	375	359
Turbo Prolog Compiler by Borland International	100	79

assembly language

8088 Assembler *w/Z-80 Translator* by 2500 AD	100	89
ASMLIB *Function Library* by BC Associates *New*	149	129
Cross Assemblers *Over 25 Varieties* from 2500 AD	CALL	CALL
Microsoft Macro Assembler *with utilities*	150	99
Turbo EDITASM *Fast Assembler* by Speedware	99	84
Visible Computer: 8088 by Software Masters	80	65

basic language

BetterBASIC by Summit Software	200	165
8087 Math Support	99	85
Btrieve Interface	99	85
C Interface	CALL	CALL
Run-time Module	250	225
Microsoft QuickBASIC Compiler *New version 2.0*	99	79
Professional BASIC by Morgan Computing	99	79
8087 Math Support	50	47
True Basic *Includes FREE BASICA Translator*	150	105
Run-time Module	500	435

blaise products

ASYNCH MANAGER *Specify for C or Pascal*	175	136
C TOOLS/TOOLS 2 Combination Package	175	149
C TOOLS	125	105
C TOOLS 2	100	84
C TOOLS PLUS *for Lattice C 3+ and Microsoft C 3+* *New*	175	139
EXEC *Program Chainer*	95	79
PASCAL TOOLS/TOOLS 2 *Combination Package*	175	139
PASCAL TOOLS	125	105
PASCAL TOOLS 2	100	84
RUNOFF *Text Formatter* *New*	50	47
TURBO ASYNCH PLUS *for Turbo Pascal* *New Version*	100	84
TURBO POWER TOOLS PLUS *for Turbo Pascal* *New Version*	100	84
VIEW MANAGER with Source *Specify for C or Pascal*	275	209

borland products

REFLEX *Data Base System*	150	119
REFLEX Workshop *New*	70	59
REFLEX and REFLEX Workshop *Combo Package* *New*	200	159
Turbo DATABASE TOOLBOX	70	54
Turbo EDITOR TOOLBOX	70	54
Turbo GAMEWORKS TOOLBOX	70	54
Turbo GRAPHIX TOOLBOX	70	54
Turbo LIGHTNING	99	75
Turbo PASCAL *with 8087 and BCD*	100	69
Turbo Prolog Compiler	100	79
Turbo TUTOR *for Turbo PASCAL*	40	32
Word Wizard *New*	70	59
Word Wizard and Turbo Lightning *Combo Package* *New*	150	129

c compilers

C-86 Optimizing Compiler	395	289
Datalight C Compiler *Small Memory Model*	60	49
Datalight Developer's Kit *with Large Memory Model*	99	79
DeSmet C *with Source Debugger*	159	145
DeSmet C *with Debugger & Large Case Option* *New*	209	193
Eco-C *Complete Development System* by Ecosoft	125	89
Lattice C Compiler from Lattice *See Lattice Section*	500	299
Let's C Compiler by Mark Williams	75	59
with csd Source Level Debugger	150	118
Microsoft C Compiler *with CodeView* *New version 4.0*	450	299
MWC-86 by Mark Williams	495	299
Wizard C Compiler *Includes Lint* by Wizard Systems	450	369

c interpreters

C-terp by Gimpel Software *Specify compiler interface*	300	239
Instant C by Rational Systems *New version*	500	379
Introducing C by Computer Innovations	125	105
Run/C by Age of Reason	150	99
Run/C Professional by Age of Reason	250	184

c utilities

Also refer to Blaise, Lattice, Microsoft, Phoenix, Polytron, SoftCraft and Xenix System V sections.

APT Application Programmer's Toolkit by Shaw American	395	339
Basic C Library by C Source	175	135
C Essentials by Essential Software	100	85
C to dBase *ISAM Manager* by Computer Innovations	150	139
c-tree *ISAM File Manager with source* by FairCom	395	329
C Utility Library by Essential Software	185	139
C Windows by Syscom	100	89
C Wings by Syscom	50	45
CI Probe *Source Level Debugger* *Limited Quantity Sale*	225	159
CI RomPac *for C-86* by Computer Innovations	195	149
dBx *dBase to C Translator* by Desktop AI *New*	350	329
dbQUERY by Raima *New*	CALL	CALL
dbVISTA *Single-User DBMS* by Raima	195	159
with Source Code	495	429
dbVISTA *Multi-User DBMS* by Raima	495	429
with Source Code	990	849
Entelekon Combo Package *All 3 items below*	200	175
C Function Library	130	115
C Windows	130	115
Superfonts for C	50	45
Essential Graphics *No Royalties* by Essential Software	250	219
Flash-up Windows by Software Bottling of NY	75	69
GraphiC *Mono version 2.2* by Scientific Endeavors	280	219
GraphiC *Color version 3.0* by Scientific Endeavors	350	299
The Greenleaf Functions by Greenleaf Software	185	135
Greenleaf Comm Library by Greenleaf Software	185	135
The HAMMER by OES Systems	195	175
MetaWINDOWS by Metagraphics	185	139
MetaWINDOWS/Plus by Metagraphics	235	199
Multi-Halo *with Royalties* by Media Cybernetics	300	219
On-line Help from Opt-Tech Data Processing	149	119
PANEL by Roundhill *Library Source Also Available*	295	229
PC Lint by Gimpel Software *New Version*	139	109
Scientific Subroutine Library for C by Peerless	175	139
Vitamin C by Creative Programming	150	139
VC Screen *Interactive Forms Designer*	99	85
Zview *with Free Updates* by Data Mgmt Consultants	245	199

cobol language

Micro Focus COBOL Workbench	4000	3599
Micro Focus Level II COBOL	1500	CALL
COGRAPHICS *New*	250	219
COMATH	200	169
FORMS-2	300	269
Level II Animator	900	CALL
Level II SOURCEWRITER	2000	CALL
Micro Focus Level II COBOL *For Novell NetWare*	2000	1799
Micro Focus Micro/SPF	175	159
Micro Focus Professional COBOL	3000	2395
Multi-user Runtime *for PC Network*	500	449
Microsoft COBOL *See Microsoft Section*	700	495
OPT-Tech Sort *Also Sorts Btrieve Files*	149	119
Realia COBOL	995	839
RM/COBOL by Ryan-McFarland	950	675
RM/COBOL 8X *ANSI 85 COBOL* by Ryan-McFarland	1250	995

debuggers & profilers

Advanced Trace-86 *with ASM Interpreter* by Morgan	175	139
CI Probe *Source Level Debugger* *Limited Quantity Sale*	225	159
Codesifter *Execution Profiler* by David Smith *New*	119	99
Codesmith-86 *Debugger* by Visual Age	145	109
Periscope I *w/Board & Switch* by Data Base Decisions	295	249
Periscope II *w/NMI Breakout Switch Only*	145	115
Periscope II-X *Software only* *Special Price thru August*	115	85

forth language

CFORTH *Native Code Application Compiler* by LMI	300	239
LMI Forth/83 Metacompiler *Specify Target Processor*	750	599
PC/Forth by Laboratory Microsystems	150	119
PC/Forth+ by Laboratory Microsystems	250	209

fortran language

ACS Time Series by Alpha Computer Service	495	429
Btrieve *ISAM File Manager* *See SoftCraft Section*	250	195
50 MORE: FORTRAN by Peerless Engineering	125	99
For-Winds by Alpha Computer Service	90	79
Forlib-Plus by Alpha Computer Service	70	55
FORTRAN Addenda by Impulse Engineering *New*	95	89
FORTRAN Addendum by Impulse Engineering *New*	165	149
Grafmatic or Plotmatic by Microcompatibles *New*	135	119
Grafmatic and Plotmatic by Microcompatibles *New*	240	219
Microsoft Fortran	350	215
Multi-Halo *with Royalties* by Media Cybernetics	300	219
PANEL *Screen Designer* by Roundhill	295	229
RM/Fortran by Ryan-McFarland	595	394
Scientific Subroutine Library by Peerless	175	139
The Statistician by Alpha Computer Service	295	259
Strings & Things by Alpha Computer Service	70	55

9/86

lattice products

	LIST	OURS
Lattice C Compiler from Lattice	500	299
with Library Source Code	900	549
C Cross Reference Generator	50	39
with Source Code	200	159
C-Food Smorgasbord *Function Library*	150	99
with Source Code	300	195
C-Sprite *Source Level Debugger*	175	139
Curses *Screen Manager*	125	98
with Source Code	250	194
dBC *dBase File Manager for C*	250	194
with Source Code	500	388
LMK *Make Facility*	195	149
RPG II Compiler *No Royalties*	750	635
SecretDisk *File Encryption Utility*	120	95
SideTalk *Resident Communications*	120	95
Text Mgmt Utilities *(GREP,DIFF,ED,WC,Extract,Build)*	120	95
TopView Toolbasket *Function Library*	250	194
with Source Code	500	388
Z-80 C Cross Compiler	500	388
with Library Source Code	1000	789

logitech products

	LIST	OURS
Logimouse *New*	99	89
Logimouse+ *New*	119	105
Logimouse+ with Paint *New*	169	149
MODULA-2/86 Compiler	89	64
with 8087	129	104
with 512K	189	148
MODULA-2 Editor	59	49
MODULA-2 Runtime Debugger	69	59
MODULA-2 Source Package	179	155
MODULA-2 Translator *TurboPascal to MODULA-2* *New*	49	45
MODULA-2 Utilities Package	49	45
MODULA-2 Window Package *New*	49	45

microsoft products

	LIST	OURS
Microsoft BASIC Interpreter for Xenix	350	279
Microsoft C Compiler *with CodeView* *New version 4.0*	450	299
Microsoft COBOL Compiler	700	495
for Xenix	995	795
Microsoft COBOL Tools *with COBOL Source Debugger*	350	209
for Xenix	450	359
Microsoft Fortran Compiler	350	215
for Xenix	695	CALL
Microsoft LISP *Common LISP*	250	189
Microsoft Macro Assembler *with utilities*	150	99
Microsoft Mouse *Bus Version*	175	149
Microsoft Mouse *Serial Version*	195	159
Microsoft muMath *Includes muSIMP*	300	195
Microsoft Pascal Compiler	300	195
for Xenix	695	CALL
Microsoft QuickBASIC Compiler *New version 2.0*	99	79
Microsoft Sort	195	149
Microsoft Windows	99	74
Microsoft Windows Developer's Kit	500	395

other languages

	LIST	OURS
CCS MUMPS *Single-User version* by MGlobal *New*	60	55
CCS MUMPS *Multi-User version* by MGlobal *New*	450	379
Janus/ADA C Pack by R&R Software	95	89
Janus/ADA D Pack by R&R Software	900	795
Methods *Smalltalk* by Digitalk	79	69
Smalltalk/V by Digitalk *New*	99	89
SNOBOL4+ by Catspaw *New*	95	84

other products

	LIST	OURS
Dan Bricklin's Demo Program by Software Garden	75	65
FASTBACK *Backup Utility* by 5th Generation Systems	179	159
Interactive EASYFLOW by Haventree Software	150	129
Quilt Computing Combo Package *Both Items Below* *New*	199	169
QMake *Program Rebuild Utility* *New*	99	84
SRMS *Software Revision Mgmt System*	125	109
Source Print by Aldebaran Laboratories	139	119
Taskview by Sunny Hill Software *New*	70	59

phoenix products

	LIST	OURS
Pfantasy Pac *(Pfinish,Pfix+,Plink+,Pmaker,Pmate,Ptel)*	1295	949
Pfinish *Performance Analyzer* *New Version*	395	259
Pfix-86 Plus *Symbolic Debugger*	395	259
PforCe *Comprehensive C Function Library*	395	289
Plink-86 *Overlay Linker*	395	259
Plink-86 Plus *Enhanced Overlay Linker*	495	359
Pmate *Macro Text Editor*	195	149
Pre-C *Lint Utility*	295	195
Ptel *Binary File Transfer Program*	195	149

polytron products

	LIST	OURS
Polytron C Beautifier	49	45
Polytron C Library I	99	79
Polytron PowerCom *Communications*	179	139
PolyLibrarian *Library Manager*	99	79
PolyLibrarian II *Library Manager*	149	119
PolyMake *UNIX-like Make Facility*	99	79
PolyOverlay *Overlay Optimizer*	99	79
PolyWindows Products *All Varieties*	CALL	CALL
PolyXREF *Complete Cross Reference Utility*	219	179
PolyXREF *Support for one language only*	129	109
PVCS *Polytron Version Control System*	395	329
PVMFM *Polytron Virtual Memory File Manager*	199	149

softcraft products

	LIST	OURS
Btrieve *ISAM File Manager with No Royalties*	250	195
Xtrieve *Query Utility for Btrieve*	195	165
Rtrieve *Report Generator for Xtrieve*	85	75
Btrieve/N *for Networks*	595	465
Xtrieve/N *Query Utility for Btrieve/N*	395	299
Rtrieve/N *Report Generator for Xtrieve/N*	175	159

text editors

	LIST	OURS
Brief from Solution Systems	195	CALL
Epsilon *Multi-tasking Emacs-like editor* by Lugaru	195	165
FirsTime for Turbo by Spruce Technology	75	69
KEDIT *Xedit-like editor* by Mansfield Software Group	125	109
PC/VI by Custom Software Systems *New*	149	129
Personal REXX by Mansfield Software Group *New*	125	109
SPF/PC by Command Technology Corp	195	165
Vedit by CompuView	150	115
Vedit Plus by CompuView	225	180
XTC *Text Editor with source* by Wendin	99	84

turbo pascal utilities

Also refer to Blaise, Borland and SoftCraft sections.

	LIST	OURS
ALICE *Turbo Pascal Interpreter* by Software Channels	95	69
FirsTime for Turbo by Spruce Technology	75	69
Flash-up Windows by Software Bottling of NY	75	69
Multi-Halo *with Royalties* by Media Cybernetics	CALL	CALL
On-line Help from Opt-Tech Data Processing	149	119
Screen Sculptor by Software Bottling of NY	125	95
Turbo EXTENDER by TurboPower Software	85	69
Turbo Professional by Sunny Hill Software	70	49
TurboPower Utilities by TurboPower Software	95	84
TurboWINDOW by MetaGraphics	80	69

wendin products

	LIST	OURS
Operating System Toolbox *Build your own OS*	99	84
PCUNIX *Operating System*	99	84
PCVMS *Operating System Similar to VAX/VMS*	99	84
XTC *Text Editor with Pascal Source Code*	99	84

xenix system v

	LIST	OURS
Complete Xenix System by SCO *All 3 items below*	1295	1099
Xenix Development System	595	529
Xenix Operating System *Specify XT or AT*	595	529
Xenix Text Processing Package	195	169

xenix languages and utilities

	LIST	OURS
APL*PLUS/UNX System *For AT Xenix* by STSC	995	795
Btrieve *ISAM File Manager* by SoftCraft	595	465
c-tree *ISAM File Manager with Source* by FairCom	395	329
dBx *dBase to C Translator w/source* by Desktop AI *New*	550	499
dbVISTA *Single and Multi User versions* by Raima	CALL	CALL
Informix by Relational Database Systems	995	839
Informix4GL by Relational Database Systems *New*	CALL	CALL
InformixSQL by Relational Database Systems *New*	995	839
Lyrix by SCO	595	489
Micro Focus Level II Compact COBOL *For AT*	1000	899
Forms-2	400	359
Level II ANIMATOR	600	539
Microsoft Languages *See Microsoft Section*		
Networks for XENIX by SCO	595	529
PANEL *Screen Designer for AT Xenix* by Roundhill	695	CALL
RM/COBOL by Ryan-McFarland	1250	995
RM/FORTRAN by Ryan-McFarland	750	599
SCO Professional *Complete Lotus clone* by SCO	795	695

VISA *Prices are subject to change without notice.* MasterCard
Ohio customers please add 5% state sales tax.

Hours: 8:30 AM to 8:00 PM EST, Monday through Friday.

U.S.
1-800-336-1166
CANADA
1-800-225-1166
OHIO AND OVERSEAS
1-216-877-3781
CUSTOMER SERVICE **216-877-1110**

programmer's connection

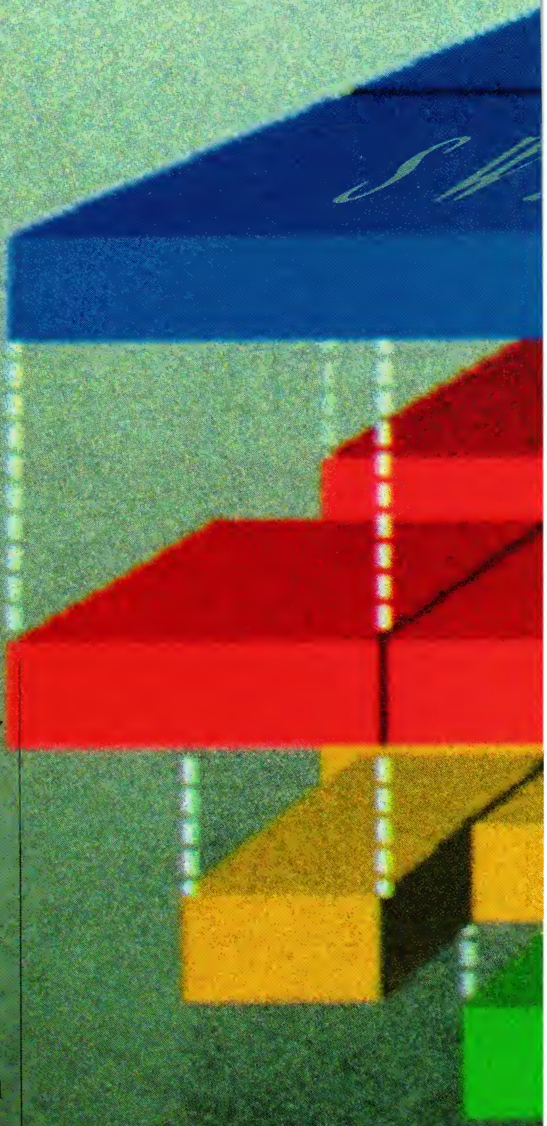
136 SUNNYSIDE ST.
HARTVILLE, OHIO 44632

CIRCLE NO. 175 ON READER SERVICE CARD

FORTH

A close look at the unusual structure of this simple language reveals the advantages inherent in a highly interactive architecture.

MAHLON G. KELLY and NICHOLAS SPIES



COMPUTER GRAPHIC • GARY ZAMCHICK/DADABASE DESIGN

Recent success in applying FORTH to networking and expert systems has generated new interest in a language traditionally associated with machine control. FORTH has always garnered attention for its unusual combination of features: a mutable environment where the lowest-level forms can be given new meanings, direct access to memory, in-line assembly language, and a highly procedural structure without the usual cost in size and speed. To appreciate its potential power, yet simple structure, requires a look at the unusual FORTH architecture. (Future articles will describe programming methods and take measure of FORTH implementations for the IBM PC.)

FORTH is both interpreted and compiled. It is an interactive language consisting of a *dictionary* of *words* that are its functions. When words (acting as commands) are typed in, they are parsed and either executed immediately (interpreted), or, if they are used in the definition of another word, compilation results. Compilation also occurs interactively at the keyboard or from source code on disk. Writing a program consists solely of defining new words, each of which may be thought of as a subroutine. The words are combined until, eventually, a final word is defined, which, when executed, executes the program. Thus, the highly structured FORTH program is simply an extension of the language itself, compiled in small pieces, each of which can be debugged as it is defined. Programs develop simultaneously top down (following a flow chart) and bottom up (by writing and testing sections).

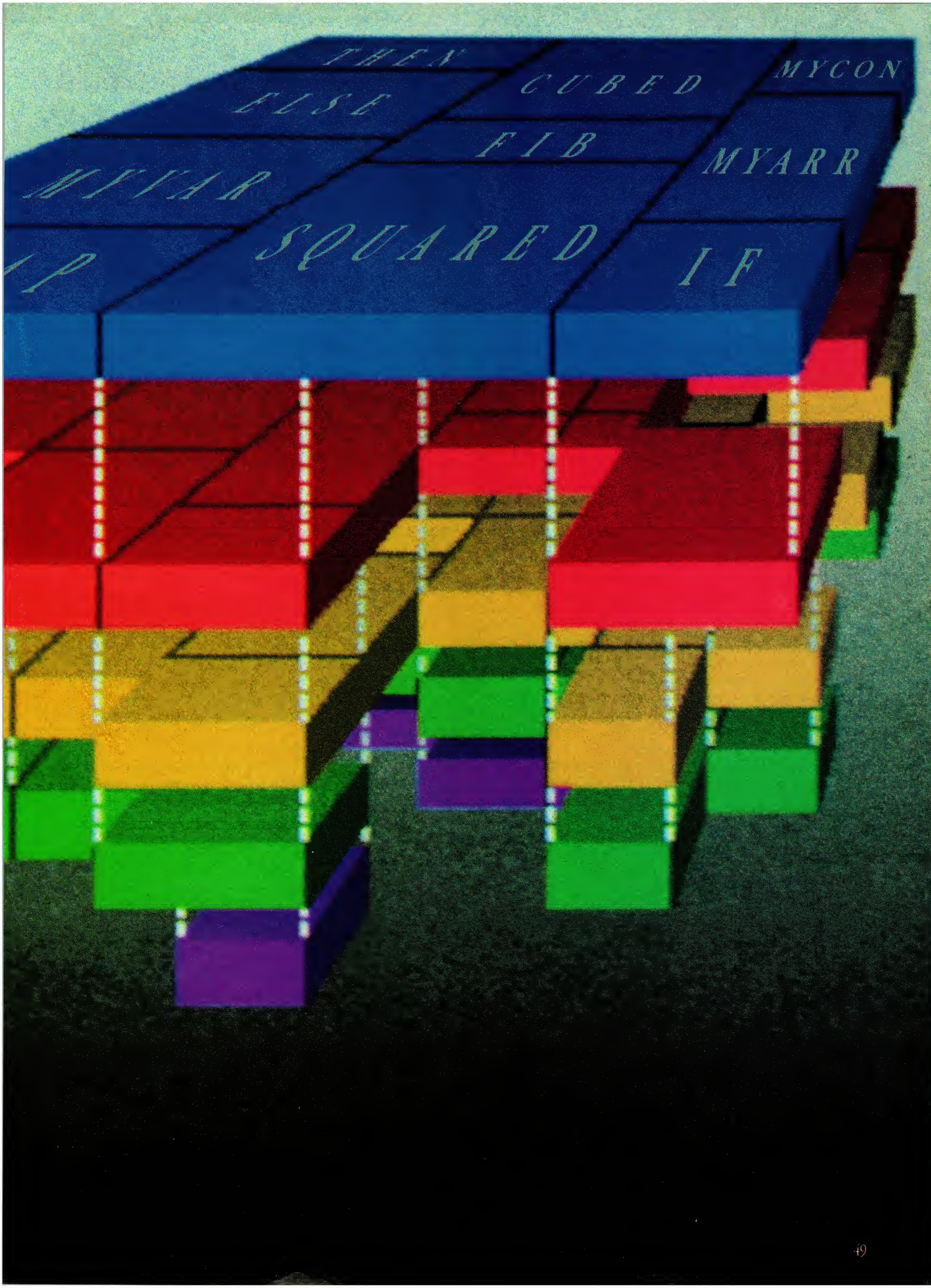
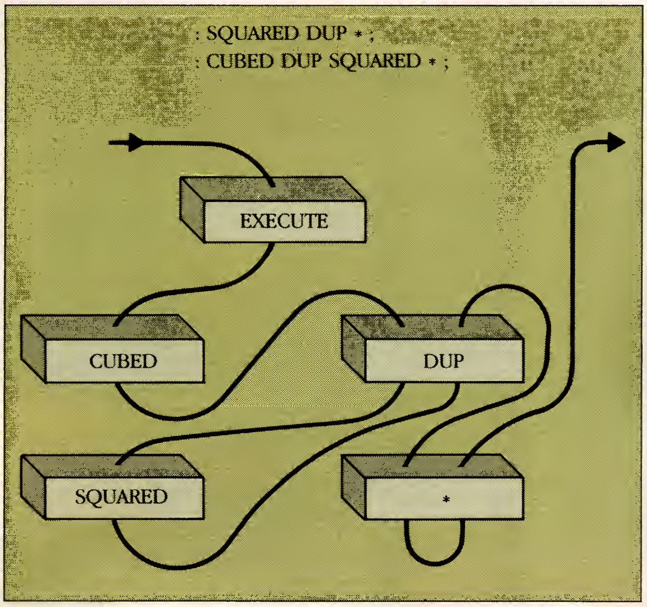
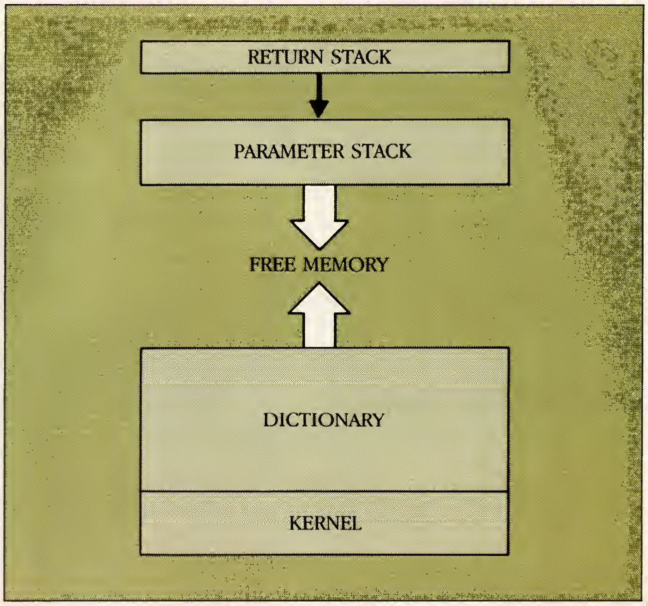

THEN
CUBED
MYCON
ELSE
FIB
MYARR
MYVAR
SQUARED
IF

FIGURE 1: *Threaded Execution*



The FORTH word CUBED holds subroutine addresses but no actual subroutine code. A single copy of the word DUP executes twice during the execution of CUBED ; the same is true for the word * (multiply). Execution by jumping sequentially to addresses is called threaded execution.

FIGURE 2: *FORTH Memory Map*



The parameter stack has a last-in-first-out operation. The dictionary holds all defined words and grows toward high memory. The kernel holds system variables, a few essential words, and other data. The return stack keeps track of the addresses to be returned to as words are executed.

Another unusual trait is the use of a last-in-first-out (LIFO) stack (generally the 8088 stack) for passing arguments among functions (named variables may be used, but their values are put on the stack before a function is executed). This use of the stack results in the application of postfix notation (also called reverse Polish notation), which radically simplifies source code and its execution. FORTH thus offers a creative environment in which programs are developed piece by piece as part of an overall problem-solving process.

The LIFO stack plays a central role in all FORTH operations. (Unless noted, all examples follow the FORTH-83 standard—see the accompanying sidebar "Building on FORTH Standards," on page 60. All numbers are expressed in decimal notation.) If the user enters

5 DUP * .

the number 25 would be displayed. The entry of 5 puts the number 5 on the stack; the next three words duplicate (DUP) the 5, multiply (*) the two numbers, and display (.) the result. [*FORTH uses punctuation symbols as commands. By convention, additional spacing is placed before and after all FORTH words throughout the text where it is necessary to separate the actual word from normal punctuation.—WF*]

FORTH parses the input stream, using spaces as a delimiter (so all words and numbers must be separated by spaces). In parsing the stream, FORTH searches the dictionary for the word. If the string is a valid word, it is executed. If it is not, FORTH determines whether the string is a valid number, in which case the number is put on the stack. If the string is neither a valid word nor number, an error message is issued.

BUILDING WITH WORDS

Compilation consists simply of defining new words (creating new dictionary entries). Each new word can be tested before the next word is added, making program development and debugging go very rapidly. Thus, if the user enters

: SQUARED DUP * ;

the word SQUARED would be added to the dictionary, having been designated so by : (colon). Entering

5 SQUARED .

then would display 25 because SQUARED executes the words used to define it. SQUARED can be used itself in subsequent definitions. Thus

: CUBED DUP SQUARED * ;

when used as

5 CUBED .

would produce (and display) 125. SQUARED and CUBED are entered into the dictionary by the action of : (colon), the most commonly used member of a special class of *defining* words that create new words. The : temporarily changes the action of the language. The first string after : is entered into the dictionary as the new word. Words that follow in the stream are not executed, but are compiled into the definition of the new word until ; (semicolon) indicates the end of the definition.

The words in a definition are compiled in an unusual way: their addresses are compiled into memory. When the new word is entered, execution jumps from address to address. Executing a program by jumping from subroutine to subroutine in primitive code as specified by a list of addresses is called *threaded execution* (see figure 1), and FORTH is called a threaded, interpreted language. The threading accounts for the compactness of FORTH compiled code; normally, all that is compiled is a sequence of addresses.

Several other defining words are available (including CREATE , described below, which is used to create new defining words). The defining word VARIABLE , when used as

VARIABLE MYVAR

creates the new word MYVAR and allots space for storage of an integer. When MYVAR is executed, the address where the number is stored is put on the stack. The word ! (called store) puts a

number that is second on the stack in the memory location that is on the top. The fetch word (@) takes an address from the stack and replaces it with a number stored at that address. Thus

```
6 MYVAR !
```

followed by

```
MYVAR @ SQUARED .
```

would display the number 36.

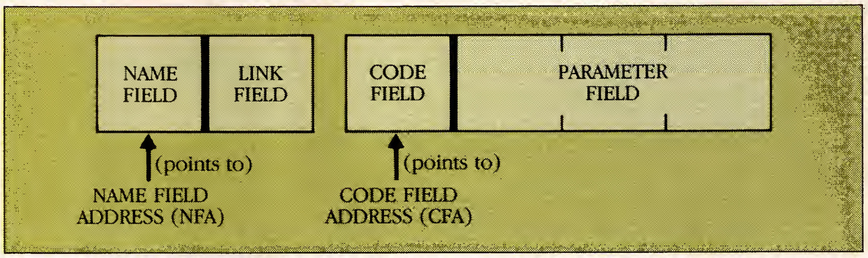
A more powerful defining word is CODE, which permits in-line assembly language definitions. CODE uses a separate dictionary of assembly language mnemonics to create new primitives. (A primitive is a machine language subroutine that executes without threading.) For example, SWAP is one of several words that manipulates the contents of the stack—it swaps the top two numbers. It could be defined as

```
CODE SWAP   BX POP   AX POP
  BX PUSH   AX PUSH   END-CODE
```

Notice that the assembler uses postfix notation. The actual words in the definition differ among dialects, as assembly language words are not standardized. This, however, illustrates a major point: FORTH is written largely in FORTH. A dialect is developed by defining a small kernel using an assembler such as MASM. After the low-level words are defined, all others are defined using FORTH itself, either with the assembler vocabulary, with colon definitions, or, rarely, by other means.

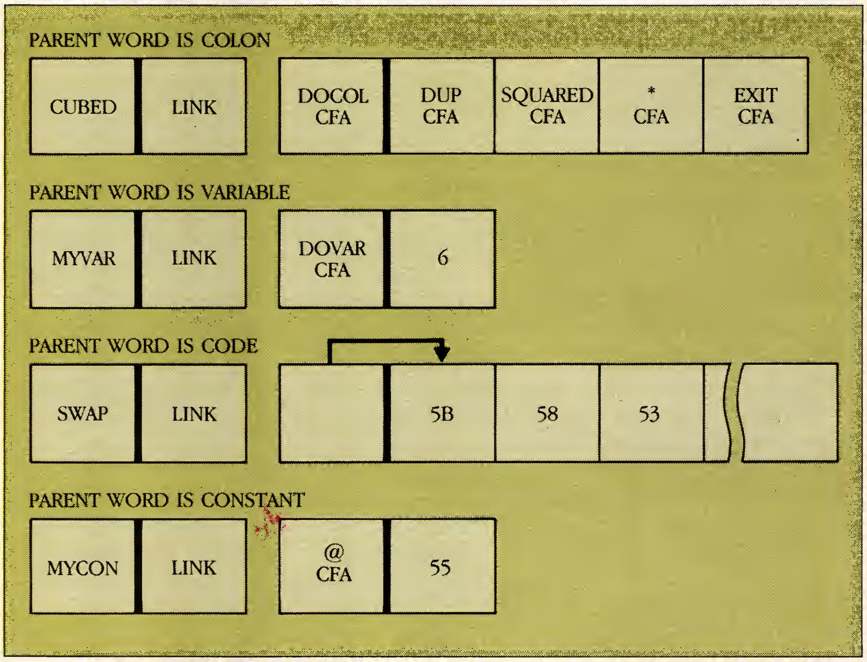
The FORTH programmer, like the user of other procedure-rich languages, such as Pascal, C, and Lisp, defines ever more powerful and general procedures (words) in terms of simpler ones that are already defined. But the FORTH user is better able to employ this style because of the minimal time and memory costs of FORTH procedure calls. FORTH invocations use a mere handful of fast and small machine-language routines, while the language's indirect addressing via threading eliminates code duplication and wasted space.

FIGURE 3: *FORTH Word Format*



The FORTH code field holds the address of the machine language subroutine for the particular class of word. The parameter field holds data to be processed by that routine, often these data are the addresses of other words to be executed.

FIGURE 4: *Various Child-word Structures*



Different defining words yield different word formats. The defining word dictates the parameter field structure and the subroutine to which the CFA points.

A DICTIONARY STRUCTURE

The structure is discussed here in terms of the simplest implementation of FORTH on the PC: a FORTH that uses 16-bit addressing within a 64KB dictionary segment, a conventional organization of the segment's memory map, and a standard scheme for threaded execution called indirect-threaded code.

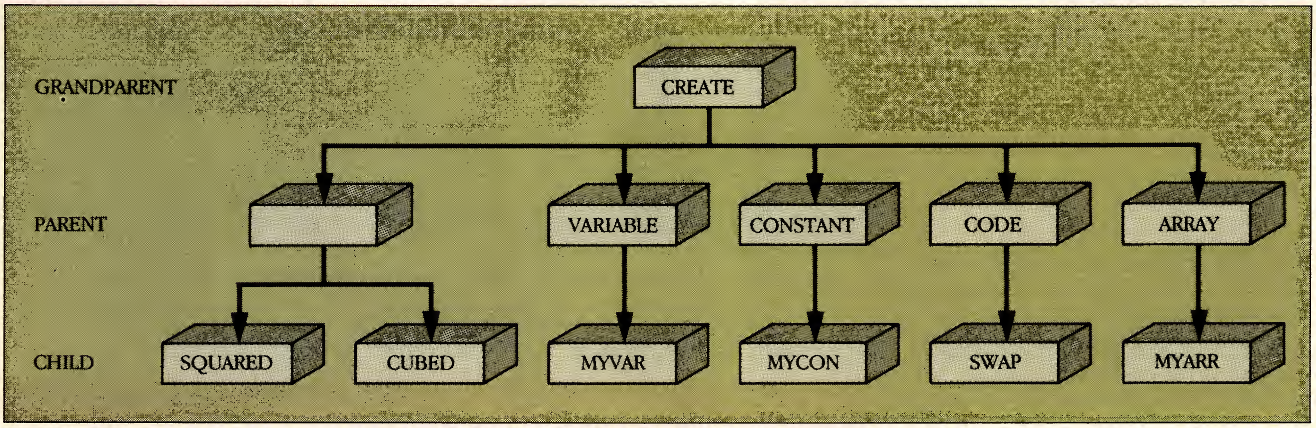
The memory map of a typical FORTH segment has four main elements (see figure 2): the dictionary, which grows upward from low memory; the parameter and return stacks, which grow downward from high memory (the parameter stack has been discussed, the return stack is used for internal purposes, as described below); a small kernel, residing below the dictionary and containing initialization code, a few machine language routines, and system tables; and two or more block buffers, a block being the basic unit of FORTH disk storage. It is the structure of the dictionary, the largest and most important part of FORTH, that determines how the language works.

The dictionary and parameter stack are at the opposite ends of free memory, permitting each to grow without interfering with the other. In a 16-bit implementation, all of these elements reside in one 64KB segment; because of this compact nature, and if data can be stored in other segments, FORTH affords sufficient space for most programs. The compiled form of FORTH words within the dictionary is a good point of departure for understanding FORTH compilation and execution.

Figure 3 shows the format in which all words are stored in memory. A compiled word consists of fours fields—name, link, code, and parameter—each located at a progressively higher address. The first two collectively are called the *header* of the definition, the second two comprise the *body*. The header lets FORTH locate and identify a word before execution or compilation into another word; the body holds the information that determines what the word will do when it is executed.

The name field contains the word's name, the identification used in diction-

FIGURE 5: *Hierarchy of FORTH Words*



CREATE controls the implicit typing in FORTH. The grandparent CREATE defines a new type, or parent word, by joining an assembly language subroutine and a parameter field structure. Each child word is a new instance of the type created.

ary searches. FORTH permits the use of any ASCII character except a space (which is used as a delimiter), a carriage return, or a line feed. Case sensitivity is dialect dependent, and in some dialects it can be switched on and off.

Next is the link field, which contains the address of the name field (NFA) of a previously defined word. This field determines the search order. When a word is referenced, a search for its definition starts at the top of the dictionary. This is a search of a linked list, from high memory down. Professional FORTH implementations often use faster, more complicated linking structures involving multiple linked lists.

The body of a word is a mini-program that contains procedure and data information. The code field points to the procedure, a machine-language subroutine called the execution primitive. The parameter field holds the data, the structure of which varies from word to word (see figure 4). The address to which the code field points—that is, the machine code primitive to be executed—depends upon the defining word used in the word's creation. Thus, the execution primitive for all words defined using : will be the same, but it will be different from the one used for words defined using VARIABLE .

The parameter field holds the data upon which the execution primitive will operate. In colon definitions, the data are actually a list of the code field addresses (CFAs) of the words used to define the new word, ending with the address of a special word called EXIT that terminates execution of the list. Compilation of a colon definition, therefore, consists of placing the CFAs of words used in the definition sequentially into the parameter field of the new word. The execution primitive manages the execution of the list of CFAs stored in the parameter field.

Figure 4 shows the contents of the earlier example word, CUBED . Its parameter field holds the CFAs of DUP , SQUARED , and * . Thus, for colon definitions, the addresses stored in the word's parameter field determine the thread of execution, and in a sense, the code pointed to in each CFA is a mini-interpreter that acts on the parameter field. By contrast, for a word created using VARIABLE, the parameter field simply contains the value stored in the variable, and the code field points to the address of a rudimentary machine language routine that returns the address of the parameter field.

The contents of the code and parameter fields in a word defined by the FORTH assembler vocabulary are noteworthy. The parameter field contains the short machine language routine that is to be executed when the word is executed. Because the contents of the code field of any word points to the machine code to be executed by the word, in the case of a CODE definition, the pointer simply points to the start of the parameter field.

GRANDPARENT, PARENT, CHILD

FORTH provides a mutable environment in which even the most fundamental forms can be redefined. A programmer who needs to tinker with the basic properties of the language must, however, understand the three-tier structure of FORTH words (see figure 5). *Child* words such as CUBED and SQUARED are the most common type of words. Because : was used to define them, it is their *parent* word, and, because they have the same parent, they are *sibling* words. Sibling words have the same execution primitive and the same parameter field structure. The primitive for : is called DOCOL (which is discussed below).

The mechanism for redefinition of fundamental forms (parent words) is CREATE , which can build new parent words. Thus, CREATE , which is a defining word, is called a *grandparent* word. When CREATE forms a new word, it builds a parameter field as specified in the word definition and links the new word to an appropriate execution primitive. The definition of the parent word VARIABLE is:

```
: VARIABLE CREATE 2 ALLOT ;
```

When VARIABLE executes to create the child word, CREATE parses the next string in the input stream and creates a header with that string (MYVAR in the earlier example) in the name field. It also creates a code field containing a pointer to a routine that returns the address of the parameter field to the stack (that routine is often called DOVAR). The 2 ALLOT allots a space of two bytes in the parameter field where the value of the variable will be stored. Now the word created with VARIABLE , and secondarily with CREATE , places the parameter field address on the stack, as did MYVAR in an earlier example. All of the children of VARIABLE are siblings; their code fields points to DOVAR .

By itself, CREATE merely specifies a parameter structure and a default child word action, DOVAR . When combined with the word DOES> , CREATE can specify the action of the child word, as illustrated by the definition of the parent word CONSTANT . CONSTANT is a defining word, the offspring of which will, instead of returning the address of

the parameter field, return the contents of that field. CONSTANT expects to find a number on the stack that it, in turn, will put into the parameter field of the new word. Thus

```
55 CONSTANT MYCON
```

defines MYCON such that when executed it puts 55 onto the stack. CONSTANT probably would be defined using CODE for speed, but it could be defined in high-level FORTH as:

```
: CONSTANT CREATE , DOES> @ ;
```

When CONSTANT is executed to create a child word, CREATE acts as it did with VARIABLE , creating a header and a code field. As a default, CREATE always points the code field to DOVAR . The word , then allots two bytes for the parameter field and puts the number from the stack there. But CONSTANT must tell the child to look up the number in the parameter field and put it on the stack. DOES> accomplishes this for CONSTANT . It specifies the runtime action of the child word, in this case to fetch the value stored in the parameter field. DOES> compiles the words that follow it into the definition of the parent word, just as if they were words in a colon definition, then puts the address of a different executable code sequence into the code field of the child word, replacing the address (DOVAR) put there by CREATE .

The new code sequence first puts the address of the parameter field on the stack (just as DOVAR did), then executes the words compiled after DOES> in the parent word as if they were in a colon definition. The net result is that MYCON first puts its parameter field address on the stack, then @ (fetches) the number stored in the parameter field to the stack. Simply, the words between CREATE and DOES> are executed when the defining word (in this case, CONSTANT) executes; the words following DOES> are executed when the child word executes.

In another example, the defining word ARRAY is redefined to create child words that are linear arrays (vectors) such that the address of an element in the array will be returned to the stack if the child word is presented with the number of the element to be returned. ARRAY should be presented with the number of elements in the array. Thus

```
10 ARRAY MYARR
```

would create the word MYARR.

```
3925 5 MYARR !
```

then would store 3925 in element 5 of the MYARR linear array, and

```
5 MYARR @ .
```

would display the contents of element 5, or 3925. The definition of ARRAY is:

```
: ARRAY CREATE 2 * ALLOT DOES>
    SWAP 2 * + ;
```

When ARRAY is executed to create MYARR , the 2 * ALLOT is executed, allotting 2 times 10, or 20, bytes (room for 10 integers) in the parameter field of MYARR . When MYARR is executed, the address of its parameter field is first put on the stack, then

```
SWAP 2 * +
```

is executed, thereby calculating the address of the element and leaving that address on the stack.

The CREATE ... DOES> construct is considered one of FORTH's more powerful features because it creates parent words that define child words with entirely new categories of action. Even more powerful, however, is the word ;CODE , which is used like DOES> , but is followed by assembly language mnemonics, permitting the action of the child word to be defined in machine language. ;CODE , which also acts with CREATE , is different from CODE (which was explained earlier).

The FORTH dictionary, then, consists of a linked list of word names. Each of the names is associated with a code field that contains a pointer to a machine code sequence that performs the action of the word; it is also associated with a parameter field that holds data of some variety that determines the exact behavior of the word.

In FORTH, interpretation has two distinct phases: location of words and management of their execution.

FORTH interpretation is responsible both for the execution of the subroutines represented by words and for the compilation of new words. If this seems paradoxical in terms of the dichotomy often drawn between interpreters and compilers in some languages, consider that in FORTH, compilation is brought about simply by the execution, through interpretation, of defining words such as : , VARIABLE , CONSTANT , and ARRAY .

THE OUTER INTERPRETER

An interpreter translates source code into an executable form and then immediately performs the actions indicated. In FORTH this process has two distinct phases—locating words in the dictionary, and managing their execution or compilation in an orderly fashion. Locating words and deciding whether to compile or execute is the responsibility of the outer interpreter; their execution is handled by the inner (or address) interpreter.

The action of the outer interpreter is, in most dialects, initiated by INTERPRET; but whether or not it is invoked by a named word, the process is similar. The operation of INTERPRET (diagrammed in figure 6) was partially revealed in the description of dictionary structure. It first parses the input stream, then looks for the parsed string in the dictionary. If it is successful, the CFA is put on the stack (this is commonly done by a FORTH word, often FIND , but the word depends upon the dialect and the FORTH standard in use).

If the string cannot be found (as indicated by FIND or its equivalent with the return of 0 to the stack), INTERPRET tests the string as a legitimate number. If the string is a number, INTERPRET converts it from ASCII to binary form on the stack (using the word NUMBER). When the string is a word or a number, the next step depends upon whether FORTH is in compile or execute mode. In either case INTERPRET is executed until the input stream is exhausted. (If the string is neither a word nor a number, the outer interpreter aborts; it ignores the remainder of the input stream, empties the stack, and generates an error message.)

The outer interpreter knows whether FORTH is in execute or compile mode by the value in a system variable called STATE . If STATE @ returns a nonzero value to the stack, then compilation occurs. If 0 is returned, FIND looks for the word. If FIND is successful, the word associated with the CFA that is returned is executed. If FIND is unsuccessful, NUMBER converts the string to a number on the stack. The outer interpreter will abort if the string is neither a word nor a number.

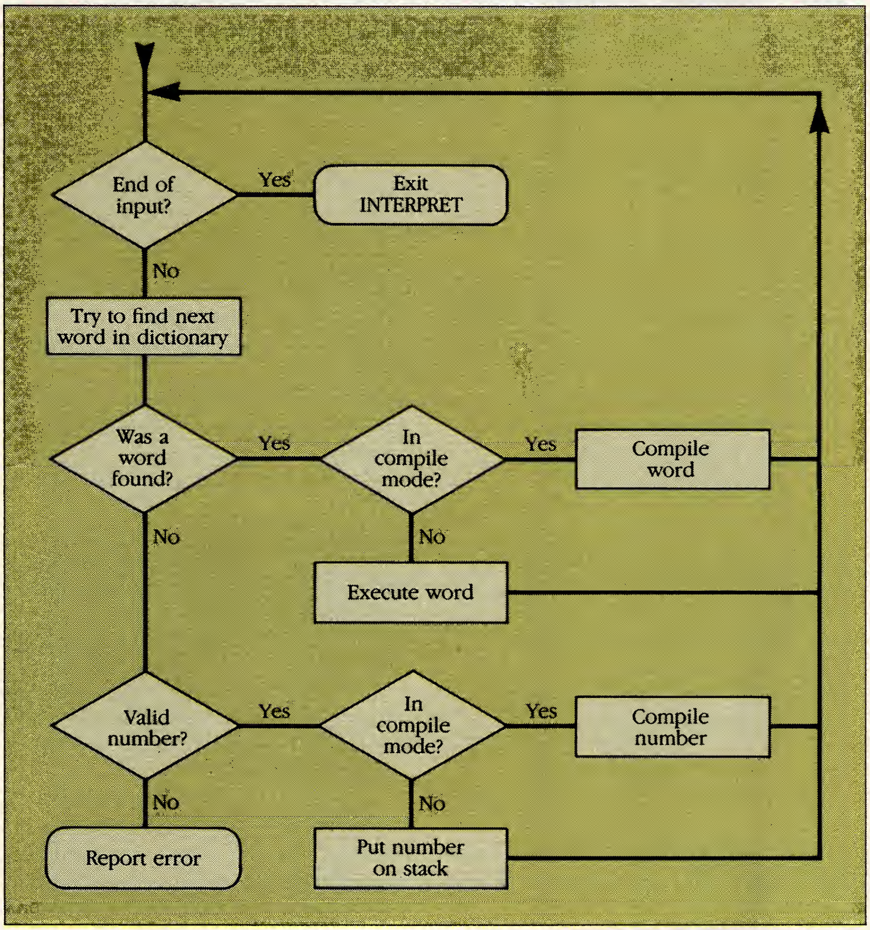
During this process, the word EXECUTE executes the word associated with the CFA at the top of the stack. In the FORTH-83 standard, the word ' (tick) returns the CFA to the stack; in FORTH-79, FIND does this job. If

```
' DUP .
```

were entered and the address 15CH

DATAFLEX®
The World's Best Selling Multi-User Database For Five Years Running.
DATAFLEX
DataFlex versions are available for all popular single-user and multi-user operating systems, including Xenix.
For a free copy of this poster (minus ad copy), write Data Access Corp. or leave a message on our Bulletin Board System: (305) 238-0640.
DATA ACCESS CORPORATION
8525 S.W. 129th Terrace–Miami–FL–33156-6565 (305) 238-0012–Telex 469021 DATA ACCESS CI
CIRCLE NO. 118 ON READER SERVICE CARD

FIGURE 6: *The Outer Interpreter*



The input stream is a series of words. The outer interpreter puts these words into a new definition in the dictionary if compile mode is on, executing otherwise.

were displayed, then entering either

5 15C EXECUTE . .

or

5 ' DUP EXECUTE . .

or

5 DUP . .

all would yield the same result. Each would display 5 5—that is, the 5 is duplicated and displayed twice. The flow of execution among the words in the definition is controlled by the inner interpreter. An example using

: SQUARED DUP * ;

and

: CUBED DUP SQUARED * ;

best illustrates how EXECUTE invokes the inner interpreter. When EXECUTE executes CUBED , it begins the trace of the thread (shown in figure 1). Execution of the primitives (DUP and *) is performed automatically by the inner interpreter and does not involve EXECUTE . The interpreter eventually threads back to INTERPRET .

FORTH keeps track of all the threading via the FORTH instruction pointer (FIP), often held in the SI register. The FIP tracks the progress of the interpreter through the current FORTH word. When that word invokes another word, the old FIP is pushed onto the return stack and the FIP is reinitialized to reflect the position in the new word.

Specifically, consider as an example the case of a colon definition. Every time a FORTH colon word is called from within another, the address of the CFA of the next word in the calling word's parameter field is stored on the return stack and execution of the called word begins. The FIP controls the progress of execution within a colon word by indicating the next element of the parameter field to process regardless of whether the called word is a colon word. To describe the threading pointers in another way, the path of execution within the parameter field is traced by the FIP. But when one colon word calls another, the FIP then must trace the contents of the parameter field in the word that is called. The location to which it must return in the calling word is put on the return stack.

Now the execution of the example can be traced. When CUBED is executed, the subroutine pointed to by the CFA is executed (as mentioned above, that machine code often is referred to as DOCOL , for do colon). DOCOL's first action is to look at the next address to be executed in INTERPRET (which is in the FIP) and put it onto the return stack. This will ensure return to the outer interpreter when execution is complete. It then uses a routine called NEXT to point FIP at the first CFA in the parameter field of CUBED, which is DUP (a primitive), and execution jumps to the address pointed to by that CFA; the CFA of DUP points to the machine code to be executed. All execution primitives end with NEXT, which increments the address in FIP by two bytes, pointing to the next CFA in the parameter field of the calling word, in this case the CFA of SQUARED.

Execution jumps again to DOCOL , this time from the CFA of SQUARED . NEXT first adds two bytes to the address in FIP and puts the resulting address on the return stack (the next CFA in CUBED to be executed when SQUARED is exited), then points FIP at the first CFA in SQUARED's parameter field, that of DUP . DUP then executes again, with NEXT incrementing FIP to * , which is executed with its NEXT aiming FIP at the last CFA in the parameter field of SQUARED .

The last CFA in every colon definition (the one put there by ;) is that of the word EXIT . This word pops the top number off the return stack, which in this case is the CFA of the next word to be executed in the parameter field of CUBED (which is *), put there by the DOCOL of SQUARED . When the EXIT of SQUARED is encountered, the top two items on the return stack contain the next CFA in the parameter field of CUBED that is to be executed, and the address to be executed next in INTERPRET. EXIT thus returns to the word (as pointed to by the return stack) that invoked the word being exited. Now the * in CUBED is executed, with NEXT pointing FIP to the EXIT of CUBED ; this then pops the address to which the thread will return in INTERPRET off the return stack, completing the execution of CUBED and returning to the outer interpreter.

This description of the mechanism seems very complex, although the pro-

LANGUAGES THAT ARE CAUSING THE BIGGEST PROGRAMMING BACKLOG IN HISTORY ARE ALSO EATING NICE BIG HOLES IN OUR POCKETS.

Whether it's BASIC, COBOL, Pascal, "C", or a data base manager, you're being held back.

Held back because the language has frustrating limitations, and the programming environment isn't intuitive enough to keep track of what you're working on.

In the real world, there's pressure to do more impressive work, in less time, and for more clients.

We've been given some incredibly powerful hardware in recent times, but the languages aren't a whole lot better than they were 20 years ago.

So, whatever language you have chosen, by now you feel it's out to get you — because it is.

Sure, no language is perfect, but you have to wonder, "Am I getting all I deserve?"

And, like money, you'll never have enough.

Pretty dismal, huh?

We thought so, too.

So we did something about it.

We call it CLARION.™

You'll call it *"incredible."*

Distributed on 7 diskettes, CLARION consists of over 200,000 lines of code, taking 3+ years to hone to "world-class" performance.

With CLARION you can write, compile, run and debug complex applications in a New York afternoon.

Even if you're in Savannah.

It gives you the power and speed to create screens, windows and reports of such richness and clarity you would never attempt them with any other language.

Because *you* would have to write the code.

With CLARION you simply design the screens using our SCREENER utility and then CLARION writes the source code AND compiles it for you in seconds.

Likewise, you can use REPORTER to create reports.

Remember, only CLARION can recompile and display a screen or report layout for modification.

And with no time wasted.

All the power and facilities you need to write great programs, faster than you ever dreamed of.

Programs that are easy to use.

Programs that are a pleasure to write.

And to you that means true satisfaction.

You've coveted those nifty pop-up help windows some major applications feature. But you can't afford the time and energy it takes to write them into your programs.

That's the way it used to be.

So we fixed that, too.

CLARION's HELPER is an interactive utility that let's you design the most effective pop-up help screens that you can imagine. And they're "context sensitive," meaning you can have help for every field in your application.

Unlike the other micro languages, CLARION provides declarations, procedures, and functions to process dates, strings, screens, reports, indexed files, DOS files and memory tables.

Imagine making source program changes with the CLARION EDITOR. A single keystroke terminates the EDITOR, loads the COMPILER, compiles the program, loads the PROCESSOR and executes the program. It's that easy!

Our data management capabilities are phenomenal. CLARION files permit any number of composite keys which are updated dynamically.

A file may have as many keys as it needs. Each key may be composed of any fields in any order. And key files are updated whenever the value of the key changes.

Like SCREENER and REPORTER, CLARION's FILER utility also has a piece of the CLARION COMPILER. To create a new file, you name the Source Module. Then you name the Statement Label of a file structure within it.

FILER will also automatically rebuild existing files to match a changed file structure. It creates a new record for every existing record, copying the existing fields and initializing new ones.

Sounds pretty complicated, huh? Not with CLARION's documentation and on-line help screens. If you are currently competent in BASIC, Pascal or "C" you can be writing CLARION applications in a day. In two days you won't believe the eloquence of your CLARION programs.

Okay, now for the best part of all. You can say it in CLARION for $295.00—plus shipping and handling. All you need is an IBM® PC, XT, AT or true compatible, with 320 KB of memory, a hard disk drive, and a parallel port.

And we'll allow a full 30 day evaluation period. If you're not satisfied with CLARION, simply return it in its original condition for a full refund.

If you're not quite ready to take advantage of this no-risk opportunity, ask for our detailed 16 page color brochure. It vividly illustrates the elegance of CLARION. Consider it a preview of programming in the fast lane.

Either way, the 800 call's a freebie.

SAY IT IN

CLARION™

A2TJ/4

1-800-354-5444

VISA · AMERICAN EXPRESS · MasterCard

BARRINGTON SYSTEMS, INC. 150 EAST SAMPLE ROAD POMPANO BEACH, FLORIDA 33064 305/785-4555

cess is actually quite simple and centers around the word NEXT. NEXT *is* the inner interpreter—all it does is increment FIP by two bytes and jump to the code pointed to by FIP before it was incremented. This can be done with as few as four bytes of machine code with the 8088 and its relatives. NEXT may increment FIP to aim at new primitives, or it may increment it to point to DOCOL at the start of a colon definition or to EXIT at its end. DOCOL and EXIT then take care of the pointers on the return stack; they control the nesting of colon definitions.

EFFICIENCY OF INTERPRETATION

To appreciate the efficiency of FORTH requires only a look at the machine code for NEXT and a few other low-level words. The definition of NEXT in MVPFORTH (Mountain View Press, Inc.), in both assembly language and machine code, is as follows:

```
AD     = LODS AX, SI
AB D8  = MOV BX, AX
FF 27  = JMP (BX)
```

The first line loads the AX register with SI's contents. The second line moves the AX register's contents to BX. Finally, the JMP line jumps to the address that is in the BX register.

In MMSFORTH (a dialect produced by Miller Microcomputer Services), the definition is

```
AD     = LODS AX,SI
93     = XCHG AX,BX
FF 27  = JMP (BX)
```

LODS loads the AX register with SI's contents. The next line swaps the AX and BX registers' contents. Then JMP jumps to the address in the BX register.

In both dialects the FIP is kept in the SI register. Both NEXTs put that address into the AX register, and the LODS instruction simultaneously increments SI as needed to point to the next CFA. The former contents of the SI register (the former contents of FIP) then goes into the BX register, and an indirect jump is made to the code at the address pointed to—that is, the code to which the FIP pointed originally. MVPFORTH uses five bytes of machine code and offers a slight edge in speed, while MMSFORTH uses four, with a minor savings in memory.

Table 1 offers another angle on the threading process that takes place in the execution of CUBED. Each word (routine) requires a number of machine-code bytes and clock cycles for execution. (Exact numbers depend on the dialect of FORTH that is used, although most are similar. These numbers are for HS/FORTH, Harvard Softworks.)

TABLE 1: *Execution Profile of CUBED*

WORD NAME	BYTES	MACHINE CYCLES
EXECUTE	3	10
DOCOL for CUBE	8	31
NEXT	4	27
DUP[a]	1	10
DOCOL for SQUARE[a]	8	31
NEXT	4	27
DUP[a]	1	10
NEXT	4	27
*(multiply)[a]	4	55
NEXT	4	27
EXIT	5	19
*(multiply)[a]	4	155
NEXT	4	27
EXIT	5	19
Total	59	575

[a]*These words are actually listed in the definitions of CUBED and SQUARED.*

Shown here are the number of machine code bytes and clock cycles required by each routine performed during the execution of the sample word, CUBED.

The words NEXT, DOCOL, EXIT, and the various primitives are all very short (and therefore fast), but they build in overhead: stepping from one machine code routine to another via NEXT and EXIT takes time. Some 49 of the 59 bytes executed are overhead; only 10 are used by DUP and *.

But the issue of bytes does not offer a fair comparison. Instead notice that only 245 of the 575 clock cycles are spent on those same overhead words; a larger portion of the machine cycles is spent performing the duplications and multiplications (which also would have to be performed in an assembly language routine). In other words, an assembly language routine written to do the work of CUBED would be 575/330 or 1.7 times faster than the definition written in FORTH. (Of course if CUBED were being executed repeatedly in a loop, it could be defined in assembly language using CODE.

Although CUBED passes through 59 bytes in the bodies of various words, only 18 bytes of body and 18 bytes of header are added to the dictionary. An assembly language subroutine would require only 14 bytes of body. If the word were defined as

```
: CU DUP DUP * * ;
```

only 16 bytes would be added to the dictionary, so assembly language is more compact here. However, in more complex—and more typical—definitions, FORTH code usually takes less space than assembled machine code.)

The speed/size profile of CUBED cannot be taken as a complete performance analysis of the language. Indeed, if the (very slow) MUL instructions are replaced with ADD instructions, changing the example to

```
: DOUBLED DUP + ;
```

which replaces SQUARED, and

```
: TRIPLED DUP DOUBLED + ;
```

which replaces CUBED, the 245 clock cycles of overhead remain because the pattern of threading has not changed. Since the function of TRIPLED can be performed by

```
MOV  AX,BX
ADD  BX,BX
ADD  BX,AX
```

(24 clock cycles), the assembly language routine written to do the work of TRIPLED would be 295/24 or 12 times faster than the definition in FORTH. Overhead in longer (and more typical) FORTH words than TRIPLED will not be considerably smaller.

The truth lies somewhere between the extremes of CUBED and TRIPLED, as revealed by the interlanguage comparison in table 2 (benchmark listings are available on PCTECHline). None of the benchmarks are optimized because the goal here is to provide representative times for standard C, assembly language, and FORTH in computation (Sieve) and function calling (Fibonacci). Optimization would proceed most rapidly in FORTH, because the programmer could either implant assembly language with CODE or employ a FORTH

optimizer, which automatically removes threading and generates direct code.

The greatest advantage of FORTH is extensibility without unnecessary cost; threading makes subroutines compact and the efficient NEXT-EXIT execution minimizes subroutine overhead. Programmers familiar with the manner in which subroutines are called in more traditional languages often feel that FORTH must be inherently slow in comparison because it seems to use so many subroutines (that is, FORTH words). This is not the case. Languages such as Pascal or FORTRAN encourage the writing of fairly bulky subroutines because such a large time overhead is involved in such tasks as saving the machine state, passing variables, or setting up areas of memory for pointers. Therefore, although they are rich in procedures and allow repeated calls to subroutines, they cannot be extensible—the cost in time is too great. Even a subroutine in assembly language carries the time overhead of the call and return instructions.

FORTH, on the other hand, offers the efficient action of EXECUTE, DOCOL, EXIT, and in particular, NEXT. Their speed is responsible for FORTH's efficient extensibility. Moreover, their use of only two bytes for each subroutine (the address) in a compiled definition results in very compact code and a rapid compilation.

TABLE 2: *Interlanguage Speed Comparison*

	ASSEMBLY	C[a]	FORTH[b]
FIBONACCI	13.24	26.37	127.44
Comparison[c]	1.0	2.0	9.6/4.8
ERATOSTHENES SIEVE	10.44	18.01	119.41
Comparison[c]	1.0	1.7	11.4/6.6

All times are in seconds and were obtained on a 640KB PCs Limited Turbo PC.
[a]The C benchmarks were taken from "The State of C" (William J. Hunt, January 1986, p. 82). The sieve times are for 20 iterations.
[b]This program was derived from the January 1983 issue of Byte *("Eratosthenes Revisited: Once More Through the Sieve," Jim Gilbreath and Gary Gilbreath).*
[c]C time is a ratio to assembly language. The FORTH times show ratio to assembly/ratio to C.
This table was prepared by Terry Colligan, president of Rational Systems, manufacturers of Instant C.

A spot check of execution speed shows FORTH to be slower than assembly language and C (times are for standard code with no attempt at optimization). But FORTH code is typically more compact and more easily optimized than C or assembly.

FORTH COMPILATION

The compile operation may be divided into two parts: the production of the header, which must be done by every defining word, and the production of the body, the details of which vary from defining word to defining word. The header portion is simple, but dialect dependent (and therefore not discussed here). The production of the code field depends upon the defining word used. A most interesting process is the compilation of the parameter field of a colon definition, which consists simply of creating a code field for the new word that contains the address of DOCOL and a parameter field that contains the CFAs of the words in the definition.

FORTH compilation always takes place at the top of the dictionary (FORTH has an incremental compiler). The address of the top of the dictionary is stored in a memory location, the contents of which are put on the stack by the word HERE. FORTH programmers often refer to HERE as being incremented, although that is not quite accurate. Actually, HERE is a word that maintains a pointer to the top of the dictionary. It is this internal pointer that is incremented. Thus

```
10 ALLOT
```

would increment HERE by 10 bytes. As mentioned above, the word , copies the number on the stack to the top of the dictionary and increments HERE by 2. It can be defined as

```
: , HERE ! 2 ALLOT ;
```

After the header of a colon definition is set up, HERE points to where the code field should go. The code field will be produced with a construct such as

```
' DOCOL 2 + ,
```

Recall that ' returns the CFA of a word, in this case the address of DOCOL. Then, 2 + calculates the address of DOCOL's parameter field (two bytes higher than the CFA). Finally, , deposits the parameter field address at HERE and increments HERE by 2. The colon word's code field now contains a pointer to the machine language routine of DOCOL. All that remains is to put the CFAs of the words used in the source-code definition of the colon word into the parameter field of its dictionary entry. This is accomplished by the outer interpreter as it parses the words of the definition in the source code.

INTERPRET checks for the string in the dictionary. If it is present, its CFA is returned to the stack. If STATE @ returns a nonzero number—if FORTH is in compile mode—then INTERPRET executes the word , and saves its CFA at HERE. This is repeated until all words in the definition are compiled as CFAs in the parameter field and until ; is encountered to compile the CFA of EXIT. (If STATE @ returns 0, FORTH is in execute mode, and INTERPRET uses EXECUTE to execute the word instead of compiling it).

To clarify the compilation process, consider the portion of INTERPRET that performs these actions. As each word in the definition is parsed, the following is executed by INTERPRET:

```
STATE @ IF , ELSE EXECUTE THEN
```

(The IF ... ELSE ... THEN construct is explained below.) When IF encounters a nonzero returned by STATE @ (FORTH is in compile mode), the word , is executed. If a 0 is returned, the EXECUTE between ELSE and THEN is performed, executing the word. Thus, the compiler for colon words is part of the definition of INTERPRET and is quite simple.

But what if a number is returned to the stack by INTERPRET—that is, the parsed string represented a number? The number must be compiled, but in a way such that it is returned to the stack on execution of the colon word—the number cannot be treated as a CFA. This is the job of LITERAL, the definition of which is:

```
: LITERAL STATE @ IF COMPILE LIT ,
  THEN ; IMMEDIATE
```

LITERAL brings up the subject of delayed execution, a property that is regulated in this definition by both IMMEDIATE and COMPILE. IMMEDIATE forces the execution of a word that would otherwise be compiled, in this case, LITERAL. Without the IMMEDIATE specifier in its definition, LITERAL would merely compile into the definition before the number. But that is not enough; LITERAL must take action at compile time to process the number. It is possible to override this immediate property with the word COMPILE, which forces the immediate word LIT to be compiled into the definition of the child word of LITERAL.

When LITERAL executes, it checks to see if STATE indicates that FORTH is

in compile mode. If it is, COMPILE LIT compiles the CFA of LIT at HERE . LIT is a primitive that fetches the two bytes following its compiled CFA (the number compiled by the word ,) to the stack and increments FIP by two, causing the compiled number to be skipped by the inner interpreter. LITERAL is an example of a *compiling word*, as are the word , and ALLOT . Thus, a number is compiled into the dictionary as four bytes; the first two are the CFA of LIT , which fetches the next two bytes—the number itself—to the stack.

The compilation of machine code or an assembly language definition is particularly illustrative of the ease with which FORTH performs powerful operations. A word that executes machine code has a normal header, but its code field contents point to the start of its parameter field, where the machine code resides. CREATE creates the header and code field, with the code field pointing to the DOVAR execution primitive; when CREATE is finished, HERE is pointing to where the parameter field will go. Another routine must shove the address of the parameter field into the code field, replacing DOVAR . The following will set up a header and code field ready to have machine code put into the parameter field:

```
: CODEHEAD CREATE HERE DUP 2- ! ;
```

HERE puts the address of the start of the parameter field on the stack, DUP 2− calculates the address of the code field, and ! puts the address of HERE (which is the parameter field address) into the code field. The parameter field is created by using the C, word, which operates the same as the word , but stores a byte, rather than an integer, at HERE . The machine code for SWAP , as defined earlier using CODE , is (in hexadecimal):

```
5B  = POP BX
58  = POP AX
53  = PUSH BX
50  = PUSH AX
```

and this must be followed by the definition of NEXT (using the MVPFORTH version, AD AB D8 FF 27 hex). Here, then, is the definition of SWAP using the defining word CODEHEAD and the machine code:

```
CODEHEAD SWAP
  5B C, 58 C, 53 C, 50 C,
  AD C, AB C, D8 C, FF C, 27 C,
```

CODEHEAD SWAP is the compile header and the CFA. The first line of machine code compiles SWAP; the second line compiles NEXT.

The definition of CODE is almost identical to that of CODEHEAD , except that it ends with a word to bring up the assembly language vocabulary. This vocabulary consists of words that perform the functions of the above numbers and C, . Each word compiles the machine instructions appropriate to the particular assembler mnemonic. Thus, the assembly language words BX POP would compile 5B as was done with 5B C, . The word END-CODE puts the code for NEXT at the end of the machine code sequence and exits the vocabulary. The vocabulary words (and their specific actions) vary from dialect to dialect, but the definition of a word in either assembly or machine language is conceptually simple. In fact, an assembler will not be necessary if the programmer is willing to compile machine instructions directly using C, .

Low-level constructs that the programmer might expect to find deep within the FORTH system actually can be defined (or redefined).

CREATING A NEW STRUCTURE

The environment presented above gives a programmer great expressive power. It is so expressive that low-level constructs the user might expect to find deep within the FORTH system actually can be defined (or redefined) by the FORTH user. Consider the FORTH structure for conditional branching, IF ... ELSE ... THEN. When IF sees a nonzero number on the stack (considered to be a "true" condition), the words between IF and ELSE are executed. If the number is 0 (false), the words between IF and ELSE are skipped and those between ELSE and THEN are executed. Execution in any case continues with the word after THEN (which, instead of providing a choice as in other languages, means, *then* go ahead with whatever follows).

The use of these words should not be confused with similar constructs in Pascal, BASIC, or C. (Consider the use of IF statements below by Pascal and C, and then by FORTH.) The ELSE is optional, but if it is missing, the words between IF and THEN are executed with a true condition on the stack; otherwise they are skipped. For example, the following pieces of code test the number on the top of the stack to see whether it is evenly divisible by 2, and then display either odd or even. In Pascal, the code would be:

```
procedure ODD_EVEN (i: integer);
begin
    if i mod 2 <> 0 then writeln('Odd')
      else writeln('Even');
end;
```

in the C language:

```
odd_even(i)
int i;
{
  (i%2) ? printf("Odd") : printf("Even");
}
```

and in FORTH:

```
: ODD/EVEN? 2 MOD IF ." Odd" ELSE
    ." Even" THEN ;
```

The 2 MOD (modulus) leaves the remainder of a division by 2 on the stack. The ." parses what follows in the input stream, up to a " delimiter, and displays the parsed string.

IF is an immediate word that, instead of being compiled (by storing the address of its code field in the dictionary), implements the branching structure. IF compiles the CFA of 0BRANCH, a primitive that does the branching. If the stack is holding 0, the action of 0BRANCH is to look at the next number in the parameter field and increment FIP so as to jump over that number of CFAs; it jumps to a point after the CFA of the word immediately preceding ELSE or THEN . If a nonzero value is on the stack, 0BRANCH skips one number and goes to the next CFA. After compiling 0BRANCH , IF puts HERE on the stack and leaves space for ELSE or THEN to put the number of CFAs to be skipped by compiling a 0, using the word , . One possible definition of IF would be as follows:

```
: IF ?COMP COMPILE 0BRANCH
    HERE 0 , ; IMMEDIATE
```

An actual definition would include more error checking. ?COMP aborts with an error message if FORTH is not in compile mode. It would be defined before IF as

```
: ?COMP STATE @ 0= ABORT" Compile
  only!" ;
```

The 0= is a comparison operator that returns a true (1 or −1 depending on the FORTH standard used) and ABORT" aborts after displaying the string up to the delimiting ". After IF executes, HERE is left on the stack and the sub-

sequent words are compiled normally until THEN, which is defined as:

```
: THEN ?COMP HERE
    OVER – SWAP ! ; IMMEDIATE
```

HERE puts the address of the top of the dictionary after compilation of the words between IF and THEN on the stack. OVER copies the second number on the stack (the address returned by the HERE in IF) to the top of the stack; the – (minus) calculates how far the top of the dictionary moved, which is the number of bytes to be skipped by 0BRANCH ; and SWAP ! stores that count at the location reserved by the 0 , in IF . THEN compiles nothing—it simply completes the action of IF .

The compilation duties of IF ... THEN may become clearer by considering the parameter field contents of a very simple word:

```
: TEST IF . THEN ;
```

The first item in the parameter field will be the CFA of 0BRANCH . That is followed by 4, the number of bytes that 0BRANCH should skip if it sees a false (it should skip the number 4 itself and then the CFA of . which is next). That is all—the only items compiled by IF and THEN were 0BRANCH and the number of bytes that 0BRANCH was to skip.

ELSE is similar to a combination of IF and THEN, but it uses BRANCH instead of 0BRANCH; its definition is

```
: ELSE ?COMP COMPILE BRANCH HERE 0 ,
SWAP HERE OVER – SWAP ! ; IMMEDIATE
```

Some dialects may use a different procedure, but the principle is the same.

This discussion illustrates several points. First, the action of the FORTH compiler and execution may be changed radically by using high-level

BUILDING ON FORTH STANDARDS

The two FORTH standards, FORTH-79 and FORTH-83, provide a *minimal* set of words that a user or commercial vendor is expected to enhance. The FORTH-83 standard tightens some ambiguities in the FORTH-79 standard, but it does not increase the power of the language and it is not upwardly compatible. Neither standard embraces string manipulation, floating-point arithmetic, file storage and recall, or DOS functions.

Transporting complex software among enhanced FORTH dialects is, therefore, very difficult—they can be so different as to seem almost like separate languages. Learning a new dialect presents the same difficulty. Selecting a dialect is perplexing, because it is difficult to determine in advance whether a dialect meets certain needs. It is even difficult to illustrate a good FORTH system without using examples from a specific dialect.

FORTH implementations for the IBM PC offer a variety of enhancements. MMSFORTH and PolyForth (FORTH, Inc.) are *native* FORTHs in that they are stand-alone systems. They do not use an external disk operating system; rather they provide the rudimentary elements themselves. Other dialects work within the context of DOS, permitting the used of DOS-based data files. Some FORTHs (PC/FORTH, from Laboratory Microsystems, Inc., and MicroMotion FORTH, for example) keep their dictionary within a single 64KB segment. This allows complete memory addressing in the dictionary with a 16-bit integer, but such dialects usually let data be stored in memory addressed with a 32-bit integer or with a paragraph address and offset (see below).

Because of FORTH's interactive nature, the language actually provides a work environment, unlike most other languages that use an editor and compiler. As part of that environment, a DOS-based FORTH that is well-tailored to the PC will provide words for direct access to many or all of the DOS interrupt and system function calls. It should rarely be necessary to exit from FORTH to DOS. A good DOS-based FORTH will provide file-manipulation commands and a shell to DOS as well. An image of an extended FORTH, as customized by the user's needs, may be saved to memory as an executable .COM or .EXE file, with, for example:

```
SAVE-EXE MYFORTH.EXE
```

A FORTH that operates under DOS should support most of the features of the operating system from within the language itself. Indeed, the boundary between FORTH and DOS may become indistinct: the user may rarely, if ever, have to leave FORTH.

The standards specify only words for working with integers, either 16-bit or, with an extension, 32-bit. The reasoning is that with proper scaling most of the functions for floating-point operations can be obtained with integers, and with greater speed and less memory. Look-up tables with interpolation, or perhaps a numeric algorithm, can be used for transcendental functions. But this is impractical for users who want to routinely use floating-point functions. However, by employing an 8087 coprocessor and using the memory available in the PC, most commercial dialects can accommodate floating-point arithmetic, and support virtually all 8087 functions. But the word names may differ, and again, portability becomes difficult.

Another issue not covered by the standards is addressing memory outside a 64KB space—beyond what can be addressed by a 16-bit number on the stack. In fact, the standards require that all addressing use 16-bit numbers. This poses two separate problems: what happens when the dictionary extends beyond a 64KB segment, and storing data outside that segment. Various solutions are available, ranging from the PC/FORTH+ solution, which uses 32-bit numbers—reasonable but it bucks the standards—to HS/FORTH, which uses separate segments for different parts of the dictionary and two 16-bit numbers, one for the paragraph address and another for the offset. Most other PC-compatible FORTHs keep the dictionary within 64KB and use 32-bit numbers or a paragraph and offset to address data such as arrays, strings, constants, and variables in extended memory.

Perhaps the greatest problem is that the standards offer no specification for the use of files, either source code or data. Native FORTHs use 1KB blocks for both source code and data, usually physically mapped onto a disk with no directory. This is not as onerous as it may seem, for a load block may be used to keep track of what is stored where. The original goal was to save disk space and avoid the memory overhead of a directory. This scheme may be used from within DOS if one or more drives or volumes is dedicated to blocks, but within DOS it is more common to use blocks within files. Other FORTHs do away with the use of blocks, and are thus nonstandard. Some FORTHs allow all three.

Although the FORTH standards say nothing about file I/O, they do specify data format. The standards expect data to be stored in blocks, which, with the specified virtual memory scheme may be very efficient,

colon words. Basic features can be added easily. Second, the nature of compilation and execution is such that constructs such as branching are fast and take little space; an IF ... THEN construct takes just four bytes to compile and has to increment only one register (containing FIP) when it executes. Third, although basic jobs are performed simply, the IF ... ELSE ... THEN syntax is quite readable in the source code. Finally, and perhaps most importantly, this type of syntax and its implementation makes FORTH a structured language. Every branch returns to what in other languages is termed the calling location; it is not possible to make undisciplined jumps in normal FORTH code. Yet the structure is not forced overtly; in a FORTH environment, GOTO is not needed. FORTH does not call for the program to explicitly declare procedures or define subroutines, because they are an inherent part of the language structure.

This description of FORTH from the inside out demonstrates its broad potential. Its fast, compact code and highly interactive structure give it great power. Advances in optimizing compilers have not rendered threading and explicit stack usage anachronistic; these structures retain their advantage for fast edit-run cycle programming. While it may not be necessary to understand FORTH at this level to be a competent programmer, certainly tapping its full power depends upon it.

Mahlon G. Kelly is an associate professor at the University of Virginia. Nicholas Spies is a film editor at Carnegie Mellon University. They are coauthors of FORTH: A Text and Reference *(Prentice-Hall, 1986); portions of this article are derived from their book.*

analogous to the use of sector buffers by DOS, but it prevents the exchange of data in files. Most PC dialects allow files of data from other languages to be read in a flexible way.

Listing 1 highlights some features provided by a fully extended FORTH tailored to the PC. It illustrates the use of data files, strings, revectoring input and output from the screen and keyboard to the file, and parsing of data from a file. Although written in HS/FORTH, these features should be available in any dialect well adapted to the PC. The program maintains a file of names, addresses, telephone numbers, and contents; it allows new entries, searches for names, and displays or deletes an entry. The first 100 bytes of the file consist of data moved to the array REC-USE ; each byte corresponds to an entry—a 1 indicates that the entry is in use and a 0 that it is free. Above that are the 202 byte records of addresses, consisting of 200 bytes for five delimited strings using | as the delimiter, and two bytes for the carriage return and line-feed marker indicating end of the record.

Four key words are used. The first, ENTER-ADR , takes 100 bytes from the start of the file, stores them in the array REC-USE , searches for the first free record (nonzero byte), changes it to one (because the record will be used), and moves the array back to disk. It queries the operator for the name, street address, and so on, and saves the items in the proper record, using random access via the DOS file pointer to find the proper file location. The items are stored as delimited strings, and the remainder of the record is padded with bytes of 255 value with no ASCII meaning.

FIND-NAME, the second key word, accepts the address of a counted string from the stack and then parses and examines the first item in each record to see if the test string occurs within it. If it does, the name and record number are printed. SHOW-ADR , the third key word, accepts a record number on the stack, calculates the offset to that record, and moves the DOS file pointer; it then parses and displays the contents of the record. Finally, DEL-ADR deletes a specified record, after showing the contents and querying the operator, by setting the appropriate byte in REC-USE to 0 and putting five consecutive delimiters (| | | | |) into the file.

The file organization is deliberately complex to illustrate FORTH flexibility. Variable record lengths are used (100 bytes for the array, 202 for the data records); fielded input is used for the array, while parsing is used for the record items; records are accessed randomly while items are parsed sequentially. Any delimiter may be used to separate items, as specified in the array LIMIT$. Virtually any file from any language may be created or read with FORTH if the proper extensions are available.

To use the address-book program, HS/FORTH must first be run by entering HSFORTH, then the proper extensions must be loaded as:

```
MFLOAD STRINGS MFLOAD ARRAYS
```

Then

```
MFLOAD ADRPROG
```

will compile the program. At this point an address file must be set up. This can be done by entering:

```
ADR-FILE MAKE-ADR-FILE
```

ADR-FILE puts the address of the string variable with a file name on the stack. MAKE-ADR-FILE creates the file and initializes its first 100 bytes to zero, as flags to indicate whether or not a record is in use. This initialized file is now ready for data entry. If

```
ENTER-ADR
```

is executed, the user is prompted to enter some piece of information. This process can be repeated for 100 entries (or more, with modification). In searching for a name, entering

```
$" atth" FIND-NAME
```

would find all names with "atth" in them and return the full name and the record number associated with it. If, for example, Matthew Brady were found in record 15, then

```
15 SHOW-ADR
```

would show his name, address, telephone number, and a comment.

```
15 DEL-ADR
```

would remove the record, freeing it for re-use by storing a zero in the REC-USE array. It should be apparent that this program is not a single program at all, but rather several programs, each executed by different words. Since the address file is opened and closed each time a word is executed (it is not open all the time), the file to be used can be changed simply by changing the string contained in the string variable ADR-FILE . That would be done by entering a command such as:

```
$" mailing.lst" ADR-FILE $!
```

As a result, all of the words would operate on that file. The equivalent statement in the source code would make such a change permanent, so the default name can be changed before compilation is performed.

—Mahlon Kelly and Nicholas Spies

LISTING 1: ADRPROG

```
( name task )                    ( use base 10 )
TASK ADRPROG                     DECIMAL

( length of record )             ( max num of records )
200 CONSTANT RECLEN              100 CONSTANT MAX-REC

( str. variable for filename )   ( initialize filename )
12 $VARIABLE ADR-FILE            $" TESTFILE.ADR" ADR-FILE $!

( variable for file handle )     ( initialize to 0 )
VARIABLE ADRHNDL                 0 ADRHNDL !

( string constant containing item delimiter )
$" |" $CONSTANT DELIM
( byte array to show if record used; 0 if record free, 1 if used )
MAX-REC CARRAY REC-USE
( initialize MAX-REC elements to zero )
0 REC-USE MAX-REC 0 FILL
255 $VARIABLE TEST$ ( string variable for search string )
\ comment after a word name shows stack contents before and after.
\ MAKE-ADR-FILE is normally used as ADR-FILE MAKE-ADR-FILE
\ and must be used at leas once to set up a file to which records
\ may be added with ENTER-ADR
: MAKE-ADR-FILE ( $adr -- )
  DUP ADR-FILE $! MAKE-OUTPUT ( store name, make file)
  ( next line vectors output to file and initializes MAX-REC number )
  ( of zeros in start of file; used to store REC-USE array )
  ( CRT vectors output back to the video screen )
  >FILE   MAX-REC 0   DO 0 EMIT LOOP   CRT   CLOSE-OUTPUT ;
\ if ADRHNDL contains less than 5 file has not been opened
: OKFILE? ( -- ) ADRHNDL @ 5 < ABORT" File not open." ;
\ returns number of first free record in file, looking at REC-USE
: FIND-FREE ( -- first-free-rec-num )
  MAX-REC 0 DO
    ( LEAVE exits loop if REC-USE entry of 0 is found, with )
    ( loop index on the stack )
    I REC-USE C@ 0=    IF I LEAVE THEN
    ( if I+1 reaches MAX-REC there is no 0 entry in REC-USE )
    MAX-REC I 1+ = ABORT" No room for record."
  LOOP ;
\ calculate offset of file pointer given record number
: FIND-PTR ( rec-num -- lo of ptr ) RECLEN 2+ * MAX-REC + ;
\ sets MS-DOS's file pointer to calculated file pointer for a record
: FIND-REC ( rec-num -- )   OKFILE?
  ( LSEEK expects lo val. of pointer, hi val. of pointer, handle )
  ( on stack, then sets MS-DOS pointer with 42 hex system function )
  ( call; DROPs needed because LSEEK returns pointer value )
  FIND-PTR 0 ADRHNDL @ LSEEK DROP DROP ;
\ puts the low value of the end of file pointer on the stack
: FIND-EOF ( -- lo of EOF ) OKFILE?
  ( LSEEK++ with 0 as lo offset, 0 as hi offset, and handle on stack )
  ( sets MS-DOS pointer to EOF and returns lo and hi value of EOF )
  0 0 ADRHNDL @ LSEEK++
  ( set pointer to SOF, drop all but lo value of EOF )
  0 0 ADRHNDL @ LSEEK DROP DROP DROP ;
\ opens for output the filename stored in ADR-FILE ; sets ADRHNDL
\ to value of handle returned by OPEN-OUTPUT
: OPEN-ADR-OUT ( -- ) ADR-FILE OPEN-OUTPUT OUTPUT @ ADRHNDL ! ;
\ if output file is open, closes it, resets ADRHNDL
: CLOSE-ADR-OUT ( -- ) OKFILE? CLOSE-OUTPUT OUTPUT @ ADRHNDL ! ;
\ same as OPEN-ADR-OUT , but for input file
: OPEN-ADR-IN ( -- ) ADR-FILE OPEN-INPUT INPUT @ ADRHNDL ! ;
\ same as CLOSE-ADR-OUT , but for input file
: CLOSE-ADR-IN ( -- ) OKFILE? CLOSE-INPUT INPUT @ ADRHNDL ! ;
\ if number of bytes put into record is greater than the
\ number in RECLEN , it is an error; ABORT
: TOOBIG? ( len of entry -- ) RECLEN > ABORT" Record full." ;
\ fetch and store item; expects number of bytes already put into
\ record on stack, asks for string input with IN$ , fetches length
\ from count byte of string, adds 1 for delimiter that will be put
\ in, ROT rotates third on stack, old record count, to top, +
\ calculates new record byte count, DUP duplicates it, TOOBIG?
\ checks for record overrun error
: @&!ITEM ( used -- new-used ) IN$ DUP C@ 1+ ROT + DUP TOOBIG?
  ( SWAP puts string addr on top of stack, >FILE $. DELIM $. CRT )
  ( vectors output to file, prints inputted string, prints end of )
  ( item marker, |, and returns output to video display )
  SWAP >FILE $. DELIM $. CRT ;
\ moves the contents of REC-USE from the file to the array
: GET-REC-USE ( -- ) OPEN-ADR-IN
```

```
  ( <FILE vectors input from file, MAX-REC GX takes first MAX-REC )
  ( bytes from the file to a string; 1+ 0 REC-USE MAX-REC CMOVE )
  ( moves the bytes into the REC-USE array; then closes file )
  <FILE MAX-REC GX KBRD 1+ 0 REC-USE MAX-REC CMOVE CLOSE-ADR-IN ;
\ moves the contents of REC-USE from the array to the file
: PUT-REC-USE ( -- ) OPEN-ADR-OUT
  ( LSEEK sets MS-DOS pointer to start of file, then )
  ( 0 REC-USE MAX-REC >FILE TYPE "types" contents to file )
  0 0 ADRHNDL @ LSEEK DROP DROP 0 REC-USE MAX-REC >FILE TYPE CRT
  CLOSE-ADR-OUT ;
\ queries user for information then puts to file
: ENTER-ADR ( -- )
  ( gets REC-USE contents from file and finds a free record number )
  GET-REC-USE FIND-FREE
  ( marks the record used by storing a 1 in the byte, puts to file )
  1 OVER REC-USE C! PUT-REC-USE
  ( open file and move MS-DOS pointer to start of record )
  OPEN-ADR-OUT  FIND-REC 0
  ( query for information and store to file )
  CR ." Name"        @&!ITEM
  CR ." Street"      @&!ITEM
  CR ." City/State"  @&!ITEM
  CR ." Phone"       @&!ITEM
  CR ." Comment"     @&!ITEM
  ( fill to end of record with 255 ASCII [ignored] and close file )
  ( "printing" a CR or carriage return to file makes EOR marker )
  >FILE RECLEN SWAP - 0 DO 255 EMIT LOOP CR CRT CLOSE-ADR-OUT ;
\ LIMIT$ is an array containing the input delimiters used by parser
\ this sets 124, ASCII for |, the item delimiter, as needed in LIMIT$
: SET-DELIM 124 LIMIT$ 1 + C! 124 LIMIT$ 2 + C! 124 LIMIT$ 3 + C! ;
\ FIND-NAME accepts a string address on the stack and searches the
\ name fields of each record to see if the string is any part of a
\ name string. It then displays the record number and name string
\ that was found..
: FIND-NAME ( $addr -- )
  ( store the search string in TEST$ ; sets | delimiter for parsing )
  TEST$ $! SET-DELIM
  ( open file, vector from file, GC , get cursor, does tab )
  ( to beyond where the byte array is stored )
  OPEN-ADR-IN <FILE MAX-REC GC
  ( start search in loop; R# @ returns record number counter. If it )
  ( is zero the EOF was found so file is closed, control is returned )
  ( to the keyboard by KBRD , and message is given )
  MAX-REC 0  DO   R# @ 0=
    IF CLOSE-ADR-IN KBRD CR ." Search complete." ABORT THEN
    ( G$ parses item using delimiter in LIMIT$ ; DUP TEST$ INSTR )
    ( returns non-zero or true if search string was in parsed string )
    G$ DUP TEST$ INSTR
    ( if string found show with $. and display record was found in )
    IF CR $. ."  Found in record " I .
    ELSE DROP ( otherwise drop the address of the parsed string )
    THEN
    NR        ( go to the next record and do again with loop )
  LOOP KBRD CLOSE-ADR-IN CR ." Search complete." ;
\ SHOW-ADR expects record number on stack and displays contents
: SHOW-ADR ( rec# -- )
  ( return a true if the record is empty and abort with message )
  DUP GET-REC-USE REC-USE C@ 0= ABORT" Empty record"
  ( set up the delimiters and open the file )
  SET-DELIM OPEN-ADR-IN
  ( return a true if the needed pointer is beyond the EOF )
  DUP FIND-PTR FIND-EOF >=
  ( and if it is close the file with an error message )
  IF DROP CLOSE-ADR-IN CR ." Beyond EOF." ABORT THEN
  ( set the MS-DOS pointer to the start of record and vector input )
  FIND-REC <FILE
  ( parse 5 items and display them, then close file )
  5 0 DO CR G$ $. LOOP KBRD CLOSE-ADR-IN ;
\ DEL-ADR displays the contents of a record and asks if it should be
\ deleted. If yes, the appropriate element in REC-USE is set to 0
\ and the record is blanked by placing 5 | delimiters with nothing
\ between them in the record
: DEL-ADR ( rec# -- ) DUP SHOW-ADR
  ( KEY gets a character from the keyboard; if it is not ASCII 89 )
  ( which is Y, the function is aborted )
  CR ." Delete it (Y/N)? " KEY DUP EMIT 89 <> IF ABORT THEN
  ( get the array, store 0 as needed put it to file, find the record )
  GET-REC-USE 0 OVER REC-USE C! PUT-REC-USE OPEN-ADR-OUT FIND-REC
  ( vector output to file and write 5 consecutive delimiters )
  >FILE 5 0 DO DELIM $. LOOP CRT
  CLOSE-ADR-OUT ;
```

RAMpage AT™-The Cure for

the AT Blues.

Doc plays sax down at Al's on Wednesday nights. That's blues night at Al's. And sometimes, if you shut your eyes and listen to that sultry, wailing music, you'd swear there's more than just one, lone sax playing. Doc knows his sax...been playing the same one over thirty years. And he plays it for all it's worth.

At AST we know the IBM® PC AT. We were the first to produce and ship an AT enhancement product. Like Doc's sax, your PC AT is a remarkable instrument with great potential. But if you're not using our RAMpage AT expanded memory solution, you're not playing it for all it's worth.

THE AST PREMIUM SERIES™

A Sure Cure For The AT Blues. Our RAMpage AT with up to 2 megabytes of RAM and DESQview™ multitasking/windowing software lets your PC AT soar to its maximum potential. And lets you realize your full AT investment value.

RAMpage AT For Multitasking Power. The RAMpage AT's powerful features combined with the PC AT's speed offer fast and functional multitasking capability.

Only with RAMpage AT can you sort a database, watch on-line stock quotations, calculate a spreadsheet, print labels and write a letter using your favorite word processing package...all at the same time...on the same PC AT. And with DESQview's windows you'll view and access up to nine applications instantly. You'll work faster and avoid time-consuming workflow interruptions.

Compatibility You Can Rely On. Our Enhanced Expanded Memory Specification is simply a more powerful, fully compatible version of the Lotus®/Intel®/Microsoft® (LIM) Expanded Memory Specification (EMS). With RAMpage AT you can use *All EMS Software.*

LOTUS COMPATIBLE Lotus ASK YOUR DEALER FOR APPROPRIATE CONFIGURATIONS

It's even certified compatible by Lotus Development for 1-2-3® Release 2 and Symphony® 1.1.

Powerful Features. RAMpage AT offers up to two megabytes of expanded memory that can also be used as extended memory, DESQview multitasking/windowing software, SuperPak™ utilities and our two-year warranty. Additionally, AST offers several models of the RAMpage AT-Pak, an optional module featuring serial, parallel and game ports.

Play Your AT For All It's Worth With RAMpage AT. For more information call our Product Information Center **(714) 863-1480.** Or send the coupon to AST Research, Inc., 2121 Alton Avenue, Irvine, California 92714-4992.

RAMpage AT and SuperPak trademarks of AST Research, Inc. IBM registered trademark of International Business Machines Corp. DESQview trademark of Quarterdeck Office Systems. Lotus, 1-2-3 and Symphony registered trademarks of Lotus Development Corp. Microsoft registered trademark of Microsoft Corp. Intel registered trademark of Intel Corp. The RAMpage AT Expanded Memory Board has successfully passed compatibility testing with 1-2-3 Release 2 and with Symphony 1.1 on an IBM AT with PC-DOS 3.1. Two megabytes of memory were installed. The Lotus logo is used by permission of Lotus Development Corp.

AST RESEARCH INC.

Yes, I want the cure for the AT blues. Please send me more information about RAMpage AT.

Name ____________________

Title ____________________

Company ____________________

Address ____________________

City ____________ State ________ Zip ________

( )

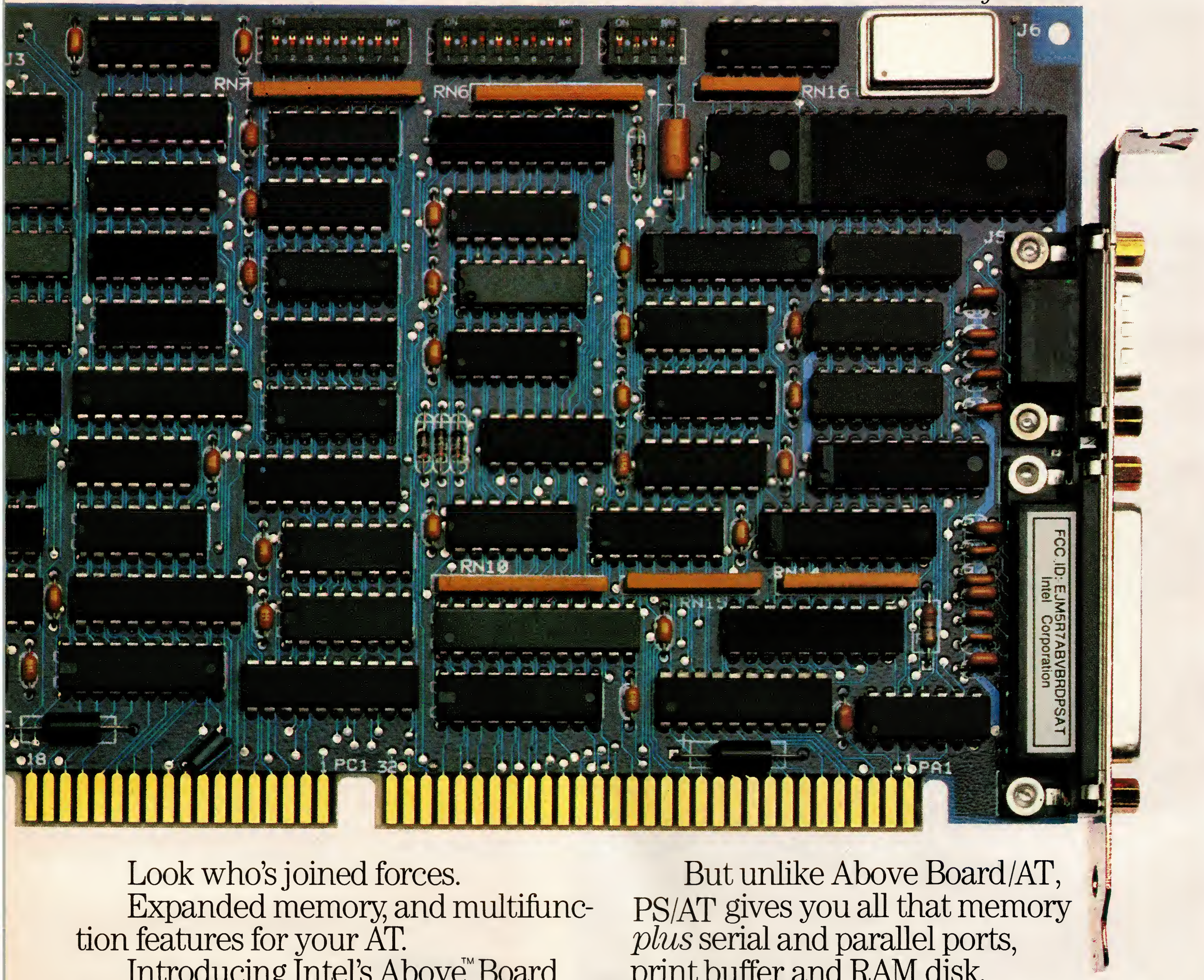
Phone ____________________

AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992.

PCTR/AT-B1

Intel announ

Multifunction



Look who's joined forces.

Expanded memory, and multifunction features for your AT.

Introducing Intel's Above™ Board PS/AT.

Like our Above Board/AT, PS/AT gives you expanded memory—which meets the Lotus®/Intel/Microsoft® spec, conventional memory—up to 640K, and extended memory—capable of supporting protected mode operating systems.

But unlike Above Board/AT, PS/AT gives you all that memory *plus* serial and parallel ports, print buffer, and RAM disk.

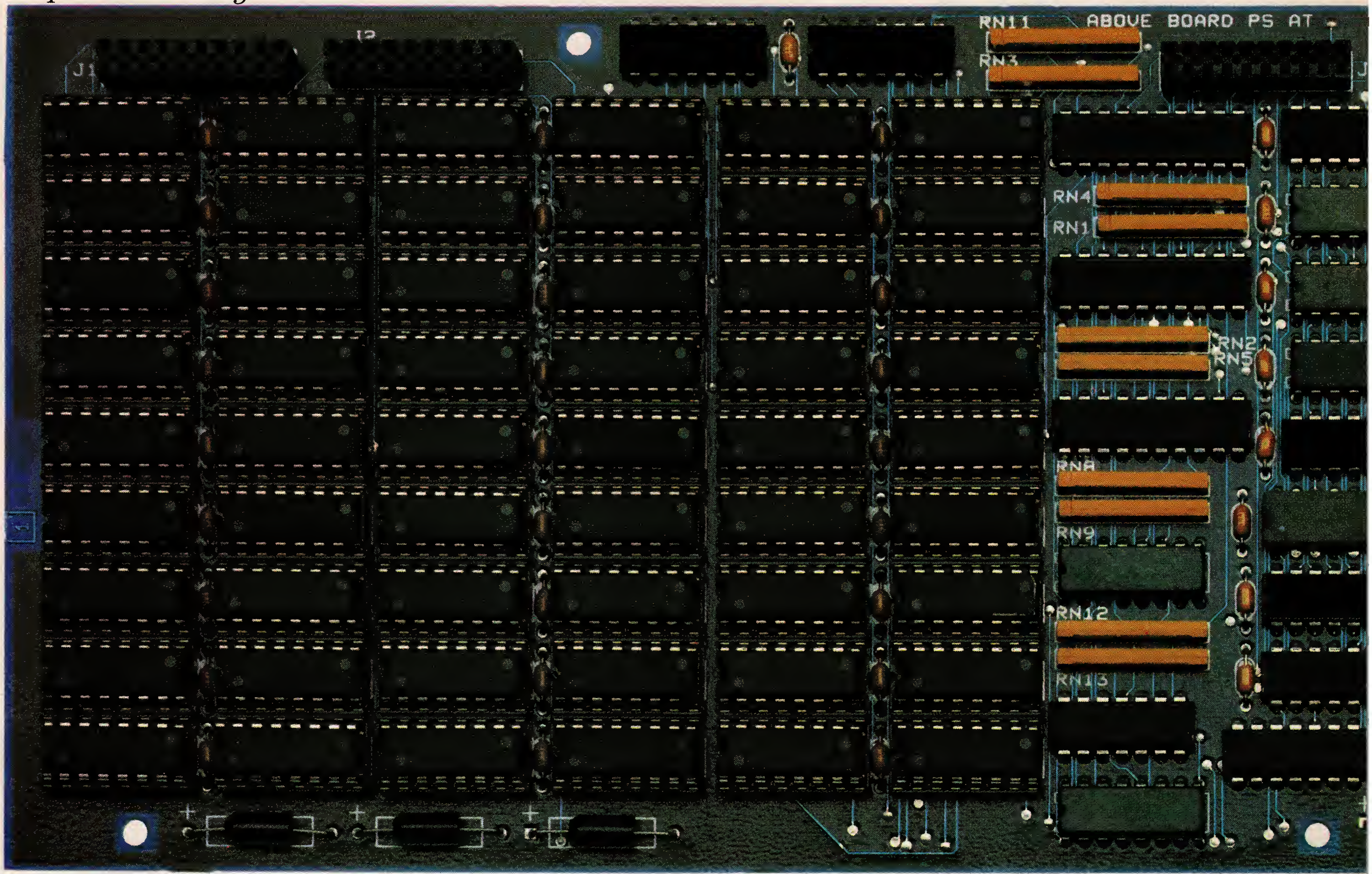
All packaged into a single slot. And all guaranteed to run at 8MHz.

PS/AT not only supplies all the memory you need, it's supported by some of the most popular software. Including 1-2-3,® Symphony,™ Framework II,™ SuperCalc,® Ready!,™ and Auto CAD.™

Lotus is a registered trademark of Lotus Development Corporation. Microsoft is a registered trademark of Microsoft Corporation. 1-2-3 is a registered trademark of Lotus Development Corporation. Symphony is a trademark of Lotus Development Corporation. Framework II is a trademark of Ashton-Tate. SuperCalc is a registered trademark of Computer Associates International. Ready! is a trademark of Living Videotext, Inc. Auto CAD is a trademark of Autodesk, Inc. Offer Valid June 23-September 30, 1986.

ces a merger.

Expanded Memory



And it's supported by somebody else, too. Us. With a free hotline, 5-year warranty—just what you'd expect from the company who cast the standard in expanded memory.

And if you're still on the fence, try this on for size:

Buy an Above Board by September 30, 1986, and we'll throw in an Intel 8087 or 80287 Math Coprocessor, free.

Plus a special offer for a free copy of Microsoft® Windows.

For details call (800) 538-3373.

So buying an Above Board PS/AT not only means one less board to buy.

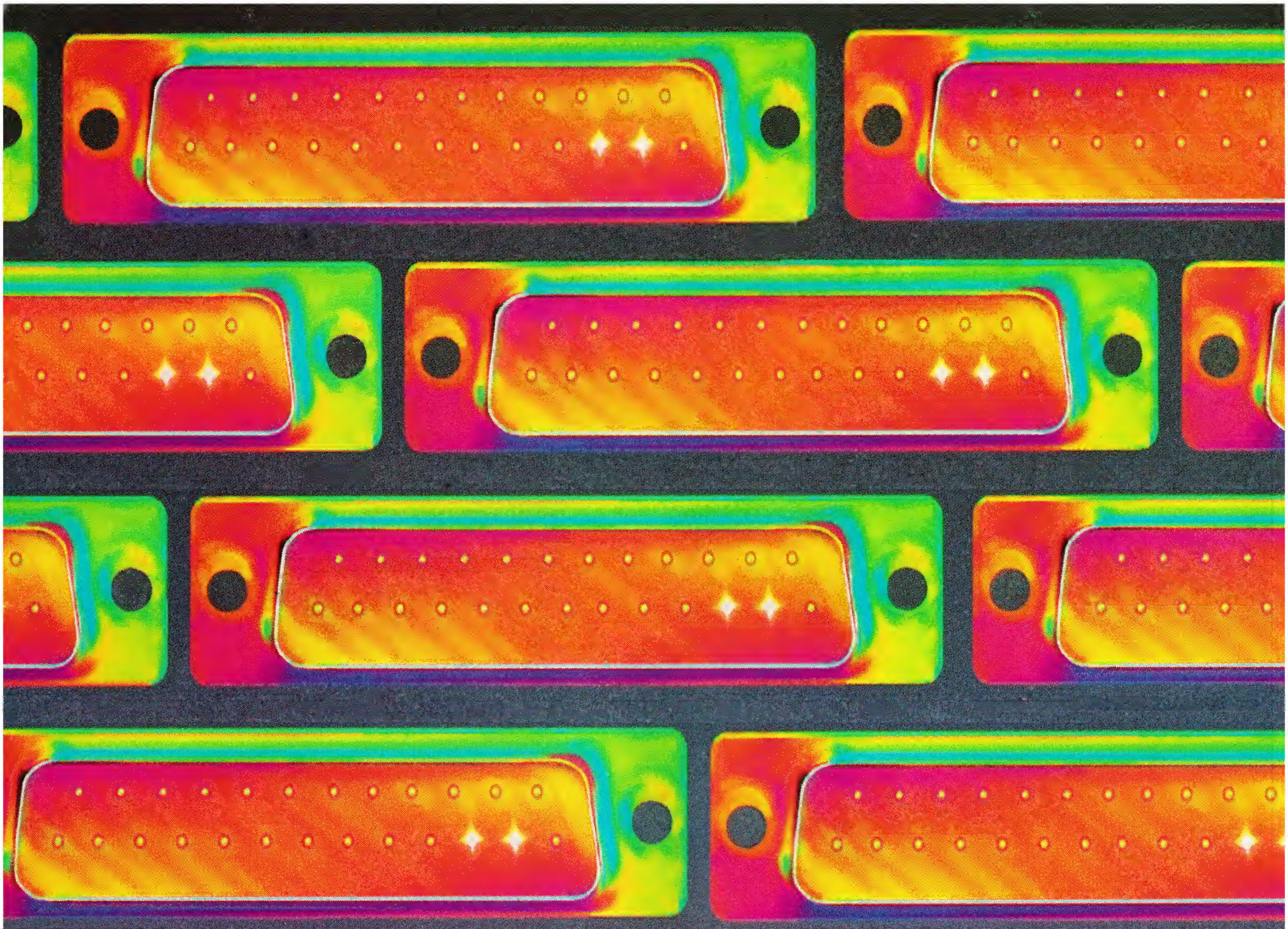
For a short time, it also means one less coprocessor to buy.

Proof that it pays to consolidate.

intel®

Good only in USA, Canada, and USA territories. Void where prohibited, taxed or restricted by law. Valid for all versions of Above Board. Redeemable only through Intel.

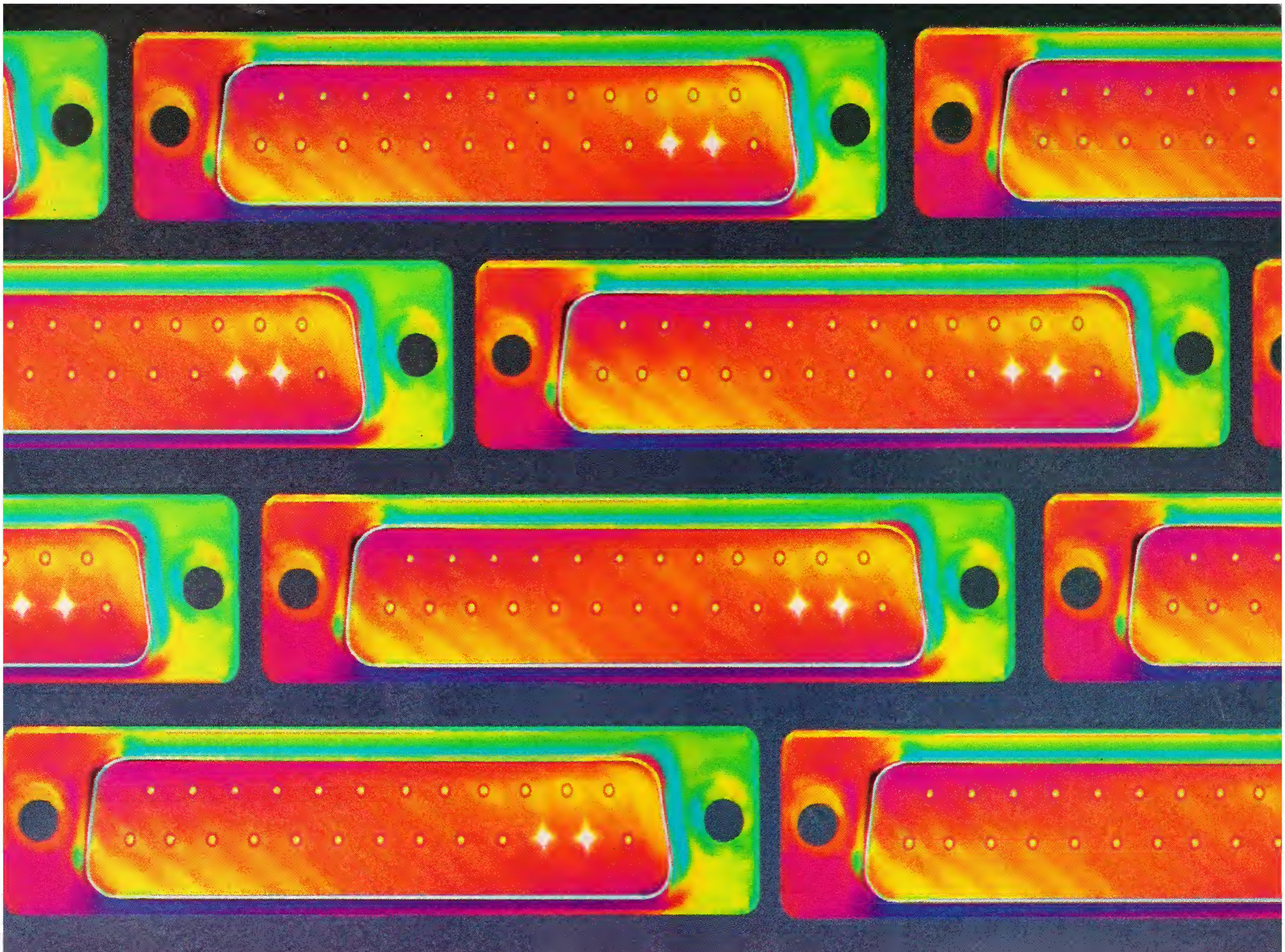
CIRCLE NO. 216 ON READER SERVICE CARD



Beyond COM2

The new multiport boards can add as many as 32 serial ports to a PC, especially with PC implementations of UNIX.

AUGIE HANSEN



In the ever-expanding world of personal computers, the two serial ports supported by the IBM PC family may no longer be sufficient. The demand for those two serial interfaces, COM1 and COM2, now extends beyond the uses of modem and serial printer.

The trend has been toward full workstation capability for PCs; peripheral devices such as serial mice, image digitizers, local area network adapters, and laser printers are all competing for the PC's two ports.

Further, an increased number of serial ports on a PC is needed to support realtime system software that collects data from several sources. Even with heavy use of signal multiplexing over a single data path, the availability of only two standard serial ports is restrictive. Multiuser operating systems running on PCs are limited to only three users: one at the computer console, and two on serially connected terminals or terminal emulators.

Given these pressures to increase the number of serial ports on the PC, several companies are offering solutions in the form of multiport boards that add as many as 32 serial ports to a PC. Most of them provide software that permits DOS, XENIX, and other operating systems to use the additional serial ports. A few require the user to program the operating system interface.

The interface to UNIX serial devices is implemented through device drivers. The Santa Cruz Operation (SCO), for example, supplies its version of Microsoft XENIX with drivers already installed for several multiport boards in predefined configurations. Owners of SCO XENIX who bought the Development System can relink the XENIX kernel with drivers to modify default configurations and to add support for still other multiport boards.

Software is often provided with hardware by the manufacturers and distributors who provide drivers for IBM XENIX 1.0. Support for IBM XENIX 2.0 is beginning to emerge as this System V-compatible update of XENIX makes its way into distribution channels.

The serial port boards reviewed in this article include American Micronics Inc.'s LAMB; AST Research's FourPort/XN; Control Systems, Inc.'s HOSTESS 8; DigiBoard Inc.'s DigiCom/8; Quadram's Quadport/AT; Sealevel Systems Inc.'s Comm +4; Sigmation's SMF/AT210 Multifunction Board; and Star Gate Technologies, Inc.'s OC4000 and OC8000 Device Drivers. Each of these boards and its characteristics is summarized in table 1.

In the material that follows, the term *port* has two meanings. From a user's perspective, a port is an external access to the machine. For a programmer a port is an access point to or from the CPU. The PC architecture allocates 1,024 I/O ports, which are also called I/O addresses. To differentiate between the two uses of the word port, the ex-

COMPUTER GRAPHIC • NATALE/EAST

ternal port is called a *serial port*, and the CPU port an *I/O address*.

An 8250 universal asynchronous receiver/transmitter (UART), or its functional equivalent, is the basis of the IBM asynchronous communications adapter. (IBM calls the special-purpose microprocessor an asynchronous communications element, or ACE.) The UART is programmed by writing data to I/O addresses in its I/O address space.

Under DOS on a PC, two blocks of I/O addresses are reserved for the serial ports COM1 and COM2 (see figure 1). The block for COM1, known also as the *primary adapter*, begins at I/O address 3F8H. Eight I/O addresses are allocated (3F8H-3FFH), with the first seven accessed externally. The last I/O address is associated with the *scratch register* that may be used internally by the UART. Another block of I/O addresses, 2F8H-2FFH, is allocated to a *secondary adapter*, COM2.

Each serial port can be programmed independently for data rate (50 to 9600 bps or more) parity setting, number of data bits (5-8) and stop bits (1, 1½, or 2), and mode (half- or full-duplex). Bit 7 of the line control register is the divisor latch access bit (DLAB) and determines which of the registers addressed at I/O ports 3F8H and 3F9H (or 2F8H and 2F9H for COM2) will be used. When DLAB is set (1), the divisor latch registers are used to program the ports speed. When it is cleared (0), the transmit/receive buffers (3F8H or 2F8H) and the interrupt enable register (3F9H or 2F9H) are used.

To get the CPU's attention, serial adapter cards can issue interrupt requests on IRQ lines in the I/O channel (see figure 2). The primary adapter is assigned to IRQ4 and the secondary adapter is assigned to IRQ3. A request to the 8259A programmable interrupt controller (PIC) is prioritized and presented to the CPU on the INTR lead. When the CPU acknowledges the interrupt (on the INTA lead), the request is mapped to interrupt INT C (COM1) or INT B (COM2) and the interrupt vector table is used to find the associated interrupt service routine.

DOS and ROM BIOS services for communications support are available through software interrupts. DOS functions 03H (read: input a character) and 04H (write: output a character) under DOS interrupt 21H are noninterrupt driven and unbuffered. The read function has no status call. The read call simply waits for an input if none has been received, thus effectively hanging the system and making the DOS functions useless in most real applications. The *DOS Technical Reference* manual itself suggests using the BIOS calls or writing a custom driver.

The BIOS routines called via interrupt 14H also offer read (service 1) and write (service 2) facilities plus the ability to initialize a comm port (service 0) and check its status (service 3). This interface also is unbuffered and not interrupt driven, so it, too, is not useful in most practical applications. The user can check the status to see whether a character has been received; this avoids the problem of the blocking reads, but frequent polling is required to check for input. Throughput is typically low and characters are likely to be lost even at modest data rates.

Programs written to interface with the serial ports usually operate through specially written device drivers or function libraries designed to overcome the limitations of the BIOS and DOS functions. Properly written drivers can provide operation at 38.4 kilobits per second (Kbps) on a standard PC and 115.2 Kbps on an AT or faster machine over short direct connections. Recent versions of MicroStuf's Crosstalk XVI can operate at these speeds. High performance requires that the software respond immediately to the interrupt generated by the UART when a character is received. The interrupt service routine must accept the character and store it safely in a buffer for the main program to read when it has time.

TABLE 1: *Multiport Board Summary*

	AMI LAMB	AST FOURPORT/XN	CONTROL SYSTEMS HOSTESS 8	DIGIBOARD DigiCom/8
Dimensions[a]	13½×4¼×½	7½×4¼×½	5½×4¼×⅝	13¼×4¼×½
Number of ports	8/board 32 max.	4/board 8 max.	8/board	8/board 32 max.
Operating systems compatibility	DOS (COM1 and COM2) IBM/XENIX 1.0 SCO XENIX V[b]	DOS (COM1 and COM2) XENIX III SCO XENIX V[b]	DOS (COM2) QNX SCO XENIX V[b]	DOS (COM1 *or* COM2) IBM/XENIX 1.0 SCO XENIX V[b]

[a] *Length, height, and thickness in inches.*
[b] *Device support built into XENIX by Santa Cruz Operation (SCO).*
[c] *The Quadport-AT board with the expansion board installed requires the width of two slots.*
[d] *Nonstandard COM3 and COM4 devices may be allocated, but DOS will not recognize them.*

FIGURE 1: *Serial Adapter I/O Addresses*

PRIMARY (COM1)	SECONDARY (COM2)		
3F8 (1016)	2F8 (760)	RX/TX/DLL	RX/TX buffer (DLAB=0) Divisor latch LSB (DLAB=1)
3F9	2F9	IER/DLM	Interrupt enable register (DLAB=0) Divisor latch MSB (DLAB=1)
3FA	2FA	IIR	Interrupt identification register
3FB	2FB	LCR	Line control register
3FC	2FC	MCR	Modem control register
3FD	2FD	LSR	Line status register
3FE	2FE	MSR	Modem status register
3FF (1023)	2FF (768)		(scratch register)

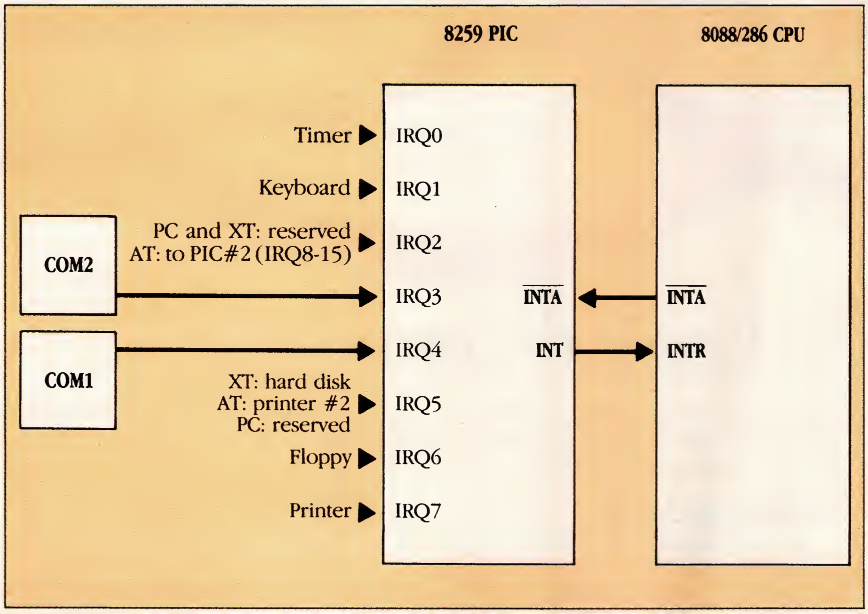
DLAB is the divisor latch access bit, bit 7 of the line control register (at 3FB or 2FB).

Each serial port is controlled and monitored via registers in the I/O address space. I/O ports for the primary and secondary asynchronous communications adapters are shown here. Other ports also may be allocated.

QUADRAM Quadport-AT	SEALEVEL SYSTEMS Comm+4	SIGMATION SMF/AT210	STAR GATE OC8000	STAR GATE OC4000
6½×4¼×1[c]	5½×4¼×½	13½×4½×⅞	13½×3⅞×½	7½×4¼×½
5/board 10 max.	4/board 8 max.	6 (with add-on board)	8/board	4/board
DOS (COM1 *or* COM2)	DOS (COM2)	DOS (COM1 and COM2)[d]	DOS (COM1 and COM2)	DOS (COM1 and COM2)

Boards without specific XENIX drivers but with sufficiently flexibleI/O address mapping can sometimes be mapped "on top of" the I/O requirements of another board's existing driver, enabling use of that driver. Remember that the Star Gate OC8000 brings only data and signal ground out to seven of its eight ports.

FIGURE 2: *Communications Interrupts*



DOS-compatible communications adapters interrupt on IRQ4 (primary) and IRQ3 (secondary) to get the CPU's attention. Some vendors of serial port boards use other interrupt levels in addition to the officially sanctioned levels.

These data rates are technically too high for RS-232, which has a nominal maximum speed of 20 Kbps, but they can be achieved over short, low capacitance lines. For example, the 38.4-Kbps speed can be used to transfer data files locally between a PC and an AT, each using standard IBM asynchronous communications adapters, with only occasional data corruption. Error-detecting protocols easily handle the corrupted data by retransmitting. The estimated throughput, allowing for the retransmissions, is approximately 32 Kbps, an improvement over 9600 bps, the stated maximum rate for the IBM asynchronous adapter. (For an introduction to PC interrupts and their applications, see "Interrupts and the IBM PC," Chris Dunford, November/December 1983, p.173 and January 1984, p. 144.)

NONSTANDARD EXTENSIONS

Some hardware and software manufacturers have extended the PC's serial interface capabilities to four separate ports. Special versions of Crosstalk supplied with internal modems are capable of using up to two additional serial ports designated COM3 and COM4 (see "Communicating from Within", Augie Hansen, September 1985, p.60). They do so by using reserved interrupts and additional I/O addresses. With an unadorned PC this is not likely to cause problems. Conflicts for interrupt requests and I/O addresses emerge, however, when these techniques are applied to newer hardware (PC/XT, PC/AT) or with older PCs that have been modified to add other devices.

The additional interrupt request levels most frequently adopted are IRQ2 and IRQ5, which were both reserved in the original PC design. The XT uses IRQ5 for the hard disk. The AT uses IRQ5 for another parallel interface and dumps the output of a second 8259 PIC onto IRQ2. This effectively adds eight new interrupt request levels (IRQ8-IRQ15) that handle the AT's hard disks, the realtime clock, and software interrupts directed to INT 0AH.

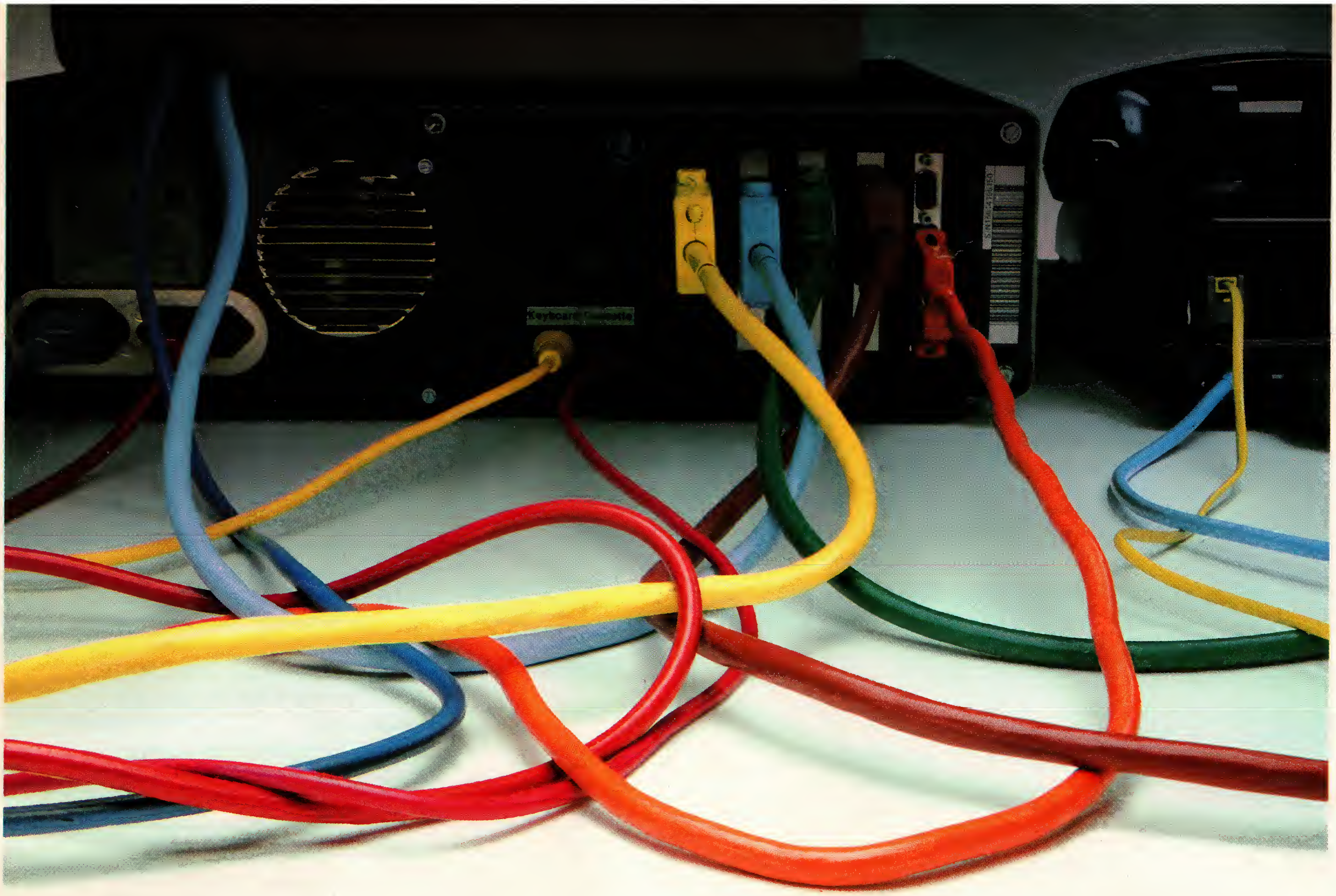
Each serial port requires its own exclusive set of I/O addresses. The addresses for the additional serial ports are usually placed in proximity to the standard COM1 and COM2 I/O addresses in areas that are marked as unused or reserved, most often in the ranges 3E8H-3EFH and 2E8H-2EFH.

The PC has 1,024 I/O port addresses (0-3FFH). Care must be taken to avoid I/O addresses used for other system functions, such as video and disk. The information for standard hardware is readily available. A more difficult problem occurs if two or more add-on board vendors have used the same range of reserved I/O addresses for their products. Unless one of the boards can be moved to new I/O addresses, *port contention* is likely to occur. It is impossible to know in advance where some unannounced product eventually is going to be I/O mapped, so a good hardware design permits switch or jumper selection of I/O addresses. This applies to the selection of interrupt request lines.

PIVOTAL SELECTION ISSUES

Two pivotal issues in configuring serial ports are the selection of interrupt levels and I/O addresses. Because only two hardware interrupt levels have been allocated to serial communications devices in the PC, certain tricks are necessary to place additional serial devices on-line. One approach has been to take over unused interrupt levels. Unfortunately, most of these already have been claimed. The original PC had several spares, but subsequent IBM machines have claimed many of them; for example, the XT hard/disk controller uses interrupt request IRQ5.

Another approach is to share an interrupt among several *cooperating devices*. This requires arbitration logic that presents one of several possibly coincident requests to the CPU in a reasonable sequence, usually based on an internal priority scheme. The common



Where's The Logical Connection™?

No more spaghetti!

If you're tired of the tangle, here's the *easy* way to connect printers, computers, modems and other devices together — in *any* configuration! Forget the old A-B switch because The Logical Connection™ is the perfect *A-Z* switch, and more! Each Logical Connection™ gives you 8 programmable ports (4 serial, 2 parallel in and 2 parallel out) that you can connect any way you want. What's more, you can "daisy-chain" them together to interconnect up to 315 separate devices. And if you have computers in more than one location, The Logical Connection™ can link everything in one office to everything in another *over a single twisted-pair wire*! (Or a single phone line with a modem at each end).

Share your printers.

Now you can give every computer in your office access to every printer — even if they're strung out all over the building (up to 4,000′ apart). And because The Logical Connection™ gives you *instantaneous* software control over which output goes to which input, you'll never have to unplug another cable to share your resources.

Fully programmable.

The Logical Connection™ provides automatic parallel to serial and serial to parallel conversion. Just plug in your devices and call up The Logical Connection's™ user-friendly configuration menu on your monitor. It will guide you step by step through the process of defining inputs and outputs, with extensive online help and easy to understand prompts. And to change connections "on the fly," you don't even have to re-configure — just precede your output with an 8-character selection string (which *you* define) and The Logical Connection™ makes the switch for you, on the spot!

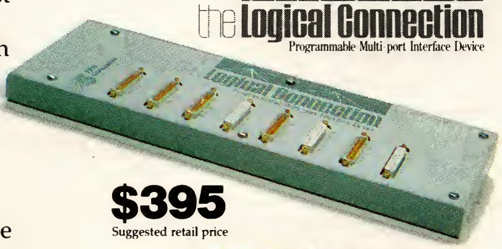
A big, smart buffer.

You won't have any slowdowns (or disasters) if three computers want to use the same printer at the same time, because The Logical Connection™ has a big 256K buffer built in (512K model also available). Smart buffer management spools each document for printing in the order received, so your computers can keep right on working.

The logical conclusion.

So if you're looking for a simple solution to the problem of connecting (and sharing) multiple printers, plotters, modems and other devices — it's time you got the connection. Get The Logical Connection™ through your leading computer dealer, or order direct from Fifth Generation Systems.

the Logical Connection
Programmable Multi-port Interface Device



$395
Suggested retail price

Fifth Generation SYSTEMS
Innovative Products Using Today's Technology

CIRCLE NO. 135 ON READER SERVICE CARD

7942 Picardy Avenue, Suite B350, Baton Rouge, Louisiana 70809
Sales: 909 Electric Avenue, Suite 308, Seal Beach, California 90740
(213) 493-4483

solution is to set flag bits in a register for each device requesting service and have a device driver tell the CPU which one to attend to first. When the operation is completed, the device bit is cleared and the next in line is serviced.

The Sigmation multifunction board allows each UART to interrupt on a separate IRQ lead. Because it is an AT-only board, a wider range of hardware interrupt levels is available than to boards that must work with the more limited range available in PCs and XTs.

Other boards use only one, two, or sometimes three interrupt levels. To maintain compatibility with DOS, many of the multiport boards allow assignment of one serial port to IRQ4 and one to IRQ3 so they can operate as standard devices COM1 and COM2. Any remaining serial devices are assigned to a common hardware interrupt.

Interrupt selection represents only half of the job of setting up a serial port. Sufficient flexibility must be built into the board to permit assigning I/O addresses to the UART registers so that they can be put in the expected places (3F8-3FF and 2F8-2FF for COM1 and COM2). Of the boards tested, most can be set to use at least one of the standard serial ports, COM1 (AUX) or COM2. A few cannot unless they are devoted exclusively to a compatibility mode, precluding the gain of having additional serial devices.

The polarity (or gender) of connectors varies among the boards tested. Most follow the pattern used by IBM. Male pins are used at the terminal equipment (computer or terminal). Most DCE devices (modems) use female connectors, permitting the use of a cable that has male pins on one end and female on the other.

TESTING THE BOARDS

All of the tested boards can be configured for at least one of the standard DOS COM ports. Some allow all ports to be configured as DOS COM ports; in those cases each port was tested individually. For many of the boards, however, only the first two ports could be configured under DOS, and for those boards only those ports were tested. None had any problem with 19.2 Kbps and all lost occasional characters at 38.4 Kbps, requiring some retransmissions during protocol file transfers, but ultimately, no information was lost.

The test set-up for DOS configurations was an AT running DOS 3.1 connected via a null-modem cable to the standard serial port on an AT&T PC6300 for hardwired link tests, and via an external Hayes-compatible U.S. Robotics Password modem for switched-network telephone link tests. Crosstalk XVI, version 3.6, running the VT-100 emulation was used to conduct tests of the hardware in a consistent way.

TABLE 2: *AMI I/O Addresses*

USAGE (system class)	**ADDRESSES**
AT	100-13F
AT	140-17F
AT	180-1BF
XT	240-27F
XT and AT	280-2BF
XT and AT	2C0-2FF
XT	300-33

Each fully configured LAMB board requires a contiguous block of 64 I/O addresses. The ranges shown above are stored in a PAL and may be changed only by replacing the PAL.

The same AT was also used to test boards while running under the Santa Cruz Operation (SCO) XENIX/286, System V, product. An additional direct serial connection was made to an XT running Crosstalk in VT-100 emulation mode. Only those boards for which SCO or the board manufacturer provides XENIX support were tested.

All of the boards that could be installed and run under SCO XENIX System V (LAMB, 4 Port/XN, HOSTESS 8, and DigiCom/8) were capable of sustained communications on several channels simultaneously at 9600 bps. The boards ran well under XENIX with two users connected via serial ports and one (the super-user) at the console.

A protocol transfer of a large file of approximately 300KB of text at 9600 bps experienced some slowing of the overall transmission rate when the XENIX scheduler triggered system activity briefly and when other users ran big jobs, such as loading the **vi** editor or searching a database. No retransmissions were required, however, and no characters were dropped because buffering was adequate on both ends, thus preventing overflow.

No discernable differences existed among the boards at the standard transmission speeds (up to 9600 bps) supported by XENIX and recommended by the board makers. This is not too surprising because all of the boards use basically the same circuit for each serial port. At higher data rates (greater than 9600 bps) a board that allows individual interrupts for each port is likely to outperform those that require polling an interrupt status register before servicing the interrupt. Because standard XENIX drivers do not support speeds greater than 9600 bps, this could not be tested in a multiuser environment.

American Micronics, Inc. The AMI LAMB (Local Area Multiuser Board) is a full-length card with eight serial ports. Four boards may be daisy-chained for a total of 32 serial ports. The LAMB employs socketed NS16450 UART chips. Two sets of four DB-25 cable-end connectors with male pins and plastic hoods come together through a slot in the board bracket to a pair of 40-pin keyed duplex headers on the board itself. The ribbon-style cables are not shielded, but have adequate grounding lines. AMI also sells 9-pin AT-compatible versions of the cable assemblies as an option.

The LAMB operates in one of two primary modes. *Split mode* assumes that no other boards in the system provide COM1 or COM2 serial ports. In the split mode, the LAMB can be set to emulate COM1 and COM2 by using the standard interrupts and I/O addresses, with any remaining serial ports on a third interrupt, or it can simply use a single interrupt for all UARTs.

In *continuous mode*, the LAMB assumes COM1 and COM2 are installed on another board and uses one of IRQ2, IRQ5, IRQ6, or IRQ7 for all UARTs. One LAMB can share an interrupt with another LAMB but not with another type of board, so several LAMBs can be stacked on the same interrupt.

Users are able to set the I/O address of an interrupt status register for each board by switches. The selected addresses for all installed boards must match those that are programmed into the device driver software.

I/O addresses are assigned to each board as contiguous blocks of 64 each. Each machine in the PC family has a core set of address assignments, but as new hardware has been added, the available addresses in the default address space of 1,024 have been diminished. AMI has programmed the address blocks shown in table 2 into a programmable array logic (PAL) device for each computer system class.

The latest release of SCO XENIX System V has support built in for the LAMB, but in continuous mode only. AMI supplies an IBM XENIX (System III) device driver. Installation requires the availability of the XENIX Software Development package. The device driver, **s8**, is provided on diskette in XENIX **tar** (tape archive) format.

The preliminary photocopied documentation provided with the evaluation LAMB is adequate in most respects. It

For $295 you can draw your own conclusions.

Drafix™ 1. The first high performance CAD software everyone can afford.

Low-cost, high performance computer-aided design & drafting (CAD) software is finally available for use on your PC.

There's never been a professional CAD package so complete at such a low cost.

Drafix 1 from Foresight Resources. Only $295.

Powerful, full-featured second generation CAD software.

Drafix 1 offers all of the capabilities you expect to find in packages costing $2,000 or more.

This is not simply a souped-up paint package. Drafix 1 is a breakthrough in design that organizes sophisticated CAD functions into smooth, fast operations. All of the drawing, designing and editing functions that designers demand from a serious CAD tool are included in Drafix 1.

A breakthrough in screen design and visual user interface.

Drafix 1 features the best organized screen design ever devised. Very simply, it displays all the information you need, all of the time.

The entire menu hierarchy is displayed constantly. There's no need to memorize commands or search for menus.

Snap-mode options are continually shown on the left screen border and can be selected "on the fly" by either pointer device or a single keystroke. Roll-down screens provide quick access to the virtually unlimited drawing, display and editing options.

And it's all controlled by a versatile three-button mouse, or digitizer, with on-screen prompts so each button function is clearly defined.

AutoCAD® compatible for easy expansion.

If you or someone in your company already uses AutoCAD, Drafix 1 offers an inexpensive way to add to your capability.

The optional Drafix 1/AutoCAD file exchange utility permits transfer of drawings between Drafix 1 and AutoCAD.

It's the perfect, low-cost alternative for increasing your drafting and design capacity.

Unbelievable low bundle prices.

To get you up and running we offer two special hardware bundles — all the equipment you need at prices you won't believe.

Get Drafix 1 with your choice of Torrington's Manager Mouse or the LOGIMOUSE® C7 for just $395.00.
Or get Drafix 1 with the SummaSketch 12″ × 12″ digitizer tablet with stylus for just $650.00.

Order today. Take advantage of our 30-day, money back guarantee.

We're so certain you'll like the performance and versatility of Drafix 1 you're welcome to try it risk free for 30 days. If you're not completely satisfied, return it to us for a full refund.

Find out just how good CAD software can be. For only $295.00.

Use our toll-free number and any major credit card to order your copy of Drafix 1 today.

SPECIFICATIONS

Code specifications
- Lattice "C" language
- Halo graphics
- Floating point database
- Coprocesser support
- PC-DOS/MS-DOS 2.1 or later

Drafix 1 Features

Items
- Lines
- Pt markers
- Arcs & circles
- Note text
- Polygons & ellipses
- Symbols/individual & nested

Item attributes
- 16 pen colors
- 255 Layers
- 8 Linetypes
- 12 Text fonts
- 32 Pt marker types

Screen Display
- Zoom/Pan/Full
- 8 Save Views
- Slide save/view
- Grids on/off

Metric & English Standards
- Engineering (decimal/fractional)
- Architectural (ft in)

Project drawing info.

Numeric input
- Keyboard and cursor
- Absolute. Relative. Polar

Snap Modes
- Gridpoint
- Endpoint
- Midpoint
- Intersect
- On item
- Quadrant
- Tangent
- Arc center

Transform/Copy
- Move
- Rotate
- Scale
- Mirror
- Align

Modify/Edit
- Attributes
- Break/Divide
- Fillet/chamfer
- Trim
- Stretch
- Erase
- Explode symbols & polygons
- Mask
- Replace
- Merge
- Region select
- Workgroup

Auto hatching, polygon fill

Auto Dimensions
- Linear Hor/Ver/Aligned
- Angular
- Dia/Radius
- Leader Notes
- Ordinate
- Chain & Baseline

Check calculations
- Coordinates
- Distance & angle
- Area & perimeter

Item Masking

Hardware

Computers
- IBM PC XT AT and compatibles
- AT&T 6300 Plus
- Zenith
- System Requirements
  - — 512K RAM
  - — RS-232 Comm port
  - — Mouse or digitizer
  - — Coprocessor recommended

Graphics Display boards
- IBM CGA/EGA and compatibles
- Hercules, and a wide range of graphics boards

Digitizers & Locators
- All popular locator devices supported

Plotters
- All popular plotters ("C" size) supported
- "E" size plotter option available

Printers
- All popular laser and dot matrix printers supported

Order Now! Call Toll-Free!

1-800-231-8574, Ext. 500

- ☐ Drafix I CAD Package $295
- ☐ Drafix I with Mouse $395
  - Torrington ____
  - LOGIMOUSE ____
- ☐ Drafix I with SummaSketch tablet $650
- ☐ AutoCAD File Exchange Utility $ 95

Dealer inquiries welcome

Check, money order, Visa and MasterCard accepted

FORESIGHT RESOURCES CORP.™
932 Massachusetts
Lawrence, KS 66044
913/841-1121

CIRCLE NO. 138 ON READER SERVICE CARD

contains good detailed information on switch and jumper settings, but does not give sufficient connection detail for the external cabling. A two-page installation procedure for the IBM XENIX software has a complete procedure for installing the **s8** device driver.

AST Research, Inc. An AT-only product (AST offers another board for PC and XT use), the FourPort/XN is a two-thirds size board that provides four serial ports and, with a suitable jumper cable, may be installed on the bus with a second FourPort/XN in order to obtain eight separate serial ports beyond COM1 and COM2.

Four cables terminate with DB-25 cable-end connectors with metal hoods, and male pins converge on a 37-pin connector to the board bracket. The high quality cables are quite sturdy and thoroughly shielded.

AST provides a 5¼-inch diskette of software that includes an IBM XENIX driver and an installation program (Qinstall) that simplifies the installation process. SCO XENIX is shipped with a driver for the product. AST does not support DOS software for DOS serial devices beyond COM1 and COM2, but the company promises such a DOS driver in the future.

The FourPort/XN operates in one of two modes. The *compatible mode* allows the first two serial ports to be configured as the standard COM1 and COM2 under DOS and mapped to XENIX as **/dev/tty00** and **/dev/tty01**. The other two serial devices are accessed as XENIX devices **tty02** and **tty03**. A second AST board may be used in tandem to add four more serial devices (**tty04-tty07**) for a total of eight serial ports available to XENIX.

Enhanced mode uses a maximum of two FourPort/XN boards to add eight serial ports to a system that already has COM1 and COM2 on some other board, for a total of ten serial ports. The PC's own COM1 and COM2 are treated as **tty00** and **tty01** and the AST ports become **tty02-tty09**.

In enhanced mode, two switch-selectable I/O address ranges (high and low) are recognized by the XENIX driver. If two FourPort/XN boards are installed, each must be assigned to a unique range to avoid port contention.

A single interrupt is used in enhanced mode. It is set to IRQ5 at the factory but may be changed to any unused interrupt request line in the range of IRQ2 through IRQ7. The I/O ports 2BF and 1BF are scratch registers that the FourPort/XN uses to store the interrupt vectors that tell the XENIX driver which UARTS on each installed board are requesting service.

In compatible mode, IRQ4 and IRQ3 are used for COM1 and COM2, respectively, per the DOS standard, and all other serial ports are ORed together on a free interrupt. IRQ5 is the default interrupt. The associated I/O addresses for serial ports 1 through 4 in enhanced and compatible modes are listed in table 3. The addresses for serial ports 3 and 4 remain the same as in enhanced mode operation.

The FourPort/XN gets high marks for its documentation, which is clearly written, complete, and accurate.

Control Systems, Inc. The half-sized HOSTESS 4 board provides four ports in its basic configuration. HOSTESS 8, the configuration tested for this review, is made up of the HOSTESS 4 with a daughterboard and an increased capacity of eight ports. The board cannot be used in slot 8 of an XT.

Two issues that must be considered in the configuration of serial ports are selecting hardware interrupt levels and assigning I/O addresses to the UART registers.

A 50-lead ribbon cable is used to bring power, data, and signal leads out of the PC to a connector block that contains a logic board as well as connectors for each group of four serial ports. Female pins are used on the connector block's connectors.

IRQ3 is the default interrupt line. The board can be set to any one interrupt in the range of IRQ2 through IRQ7. A mask register is used to enable and disable individual ports. HOSTESS 8 is the only board reviewed that allows individual ports to be enabled and disabled from software.

I/O addresses are set by switches. Contiguous blocks of 32 (four ports) or 64 (eight ports) I/O addresses are used, with the switches setting the base address of the range. Compatibility with COM2 is obtained by using IRQ3 and a base address of 2C0H, which permits existing software to access the eighth port as if it were COM2.

The HOSTESS 8 is the only board reviewed whose I/O addresses can be mapped anywhere in the 8088's address space, rather than in one place or in a limited number of predefined ranges. It is easier to avoid I/O contention with the HOSTESS 8 than with the others. This feature also makes it easier to map the HOSTESS 8's ports onto I/O address assignments of existing serial device drivers written for other boards incorporating an 8250B or INS16450 UART, allowing the HOSTESS 8 to use device drivers meant for other boards. The HOSTESS 8 supports QNX in this fashion, when mapped to a base address of 280H.

SCO XENIX System V is shipped with a driver for the HOSTESS 8. No driver software is sold with the board itself. An *Owner's Instruction Manual* and a *User's Guide* are provided with the board. These documents do not agree on the top speed of the board; the former claims 9600 bps and the latter, 19,200 bps. Few operating system serial drivers support speeds greater than 9600 bps, and yet most ports can be made to run reliably at 19,200 bps; this may have been the cause of the confusion in the documentation.

DigiBoard. The DigiCom/8 is a full-length board that supports eight serial ports. Up to four boards of similar type may be daisy-chained to get 32 separate ports. A four-port version, the DigiCom/4, is also available from DigiBoard.

One four-foot shielded cable for each of the eight ports converges on a 78-pin high-density connector on the mounting bracket. Each cable ends in a DB25 male connector with plastic hood.

The interrupt structure is an interesting design. Jumpers select one even and one odd numbered interrupt line to be distributed to the various UARTs. Each UART can be individually attached to the even or the odd interrupt line in the IRQ2 through IRQ7 range. The factory configuration has all UARTs preset to IRQ3, and the even interrupt (IRQ4) is not used. By careful selection of the I/O addresses, either COM1 or COM2 can be emulated.

When used in an AT, the preset I/O address range is 100-13FH. A pair of companion interrupt status registers are placed at 140 (even interrupt) and 141 (odd interrupt). The address range for the PC and XT is preset to 2C0H-2FFH, and the interrupt status registers are at 200H and 201H. This potentially conflicts with a game control adapter if one is installed. A device driver program reads the status ports to determine which UART is requesting service and which board it is on. The status ports can be disabled effectively to shut off selected serial ports.

Without additional software, one of the eight ports on the DigiCom/8 can act as either COM1 or COM2; however, it is not possible to configure two of the ports as COM1 and COM2 at the same time. The DigiCom/8 is provided with device drivers for IBM XENIX (System III) and SCO XENIX System V. Programs included with the DigiCom/8 are COMS.SYS, a DOS device driver for up to 10 COM ports, and COMSET.EXE, a menu-driven program used to initialize individual serial ports. Other programs demonstrate the use of the DigiCom board to receive files over a communications line and to run as a dumb terminal over any of the ports (but only one at a time). Files for XENIX are in **tar** format and instructions on installation are complete and easy to follow. The DigiCom/8 documentation is plain, but it is well organized and clearly written.

Quadram Corporation. The *Quad* part of the QuadPort-AT name is related to the manufacturer's name, not to the number of ports supported. In its basic form, the QuadPort-AT offers capabilities that are essentially the same as those of the AT Parallel/Serial Adapter—one parallel interface and one serial interface on a stand-alone board. An expansion module, called a *daughtercard* in Quadram's parlance, adds four serial ports to the package for a total of five serial and one parallel. Two QuadPort-AT boards may be used simultaneously in a system.

Connections are made via a 9-pin AT-style connector on the board bracket and via four additional 9-pin extension connectors on a small housing hung from the top edge of the AT's back panel. All connectors use male pins.

The QuadPort-AT can be run in a *DOS-compatible* mode and in a *non-DOS-compatible mode*. In the DOS-compatible mode, interrupts and I/O addresses are set to the values used by either COM1 or COM2. The board functions as its IBM equivalent. The mothercard can be placed in the non-DOS-compatible mode by selecting other interrupts and I/O addresses. No DOS drivers are provided for DOS serial devices beyond COM2.

If a daughtercard is added, the whole assembly must be run in the non-DOS-compatible mode to access the daughtercard's serial ports. If the first serial port is set for COM1 or COM2 use, the remaining ports on the daughtercard cannot generate interrupts, so they are turned off.

In non-DOS-compatible mode, serial ports are named port *A* through *E* on the first QuadPort-AT board (I/O map 1). If a second board is installed, its serial ports are called ports *F* through *K* (I/O map 2.) *J* is skipped inexplicably. I/O address maps 1 and 2 are shown in table 4.

Interrupts may be assigned to either IRQ3 or IRQ4. The CPU is informed of which UART is requesting service by the bits in interrupt request register at 2D3H (or 2DBH).

A BIOS-extension driver is provided on diskette. It is designed to be used with DOS 3.0. A file called QUADPORT.SYS replaces the ROM BIOS routines accessed through BIOS interrupt 14H, services 0-3 (AH=0 to AH=3) and in the bargain, adds several useful new services: check buffer status (AH=4), purge receive buffer (AH=5), purge transmit buffer (AH=6), return serial configuration (AH=7), and return port configuration (AH=8).

No XENIX support for the QuadPort-AT exists at this time.

The documentation is a bit cryptic about some topics. The reader is forced to dig around in the details of Appendix A to find information about interrupts that should be in the main chapters. A handy "Read this first" section takes care of most of the important set-up and installation instructions.

Sealevel Systems, Inc. The Comm +4 board consists of four serial ports on a half-sized board. The board cannot be used in slot 8 of an XT. Two boards may be used in addition to standard COM1 and COM2 serial adapters on some other boards in the system. A connector housing that hangs off the back panel of a PC contains four DB-25 connectors that have male pins.

A set of switches on address lines A5-A9 is used to select the base address of a block of 32 consecutive I/O addresses. Some addresses that can be selected by the switches could conflict with other allocations. For example, a monochrome display system uses addresses in the 3A0H-3BFH range. The use of contiguous address blocks makes finding a suitable location for the board in a PC's I/O address space a bit of a challenge. Sealevel Systems suggests a few safe base addresses that should work in most systems.

The COMMBIOS software provided with the Comm +4 is quite useful. Unfortunately, the software requires that a second board has its I/O addresses contiguous with those assigned to the

TABLE 3: *AST 4 Port/XN I/O Addresses*

RANGE	PORT	ADDRESS Compatibility Mode	Enhanced Mode
High[a]	1	3F8-3FF	2A0-2A7
	2	2F8-2FF	2A8-2AF
	3	2B0-2B7	2B0-2B7
	4	2B8-2BE	2B8-2BE
Low[b]	1	3F8-3FF	1A0-1A7
	2	2F8-2FF	1A8-1AF
	3	1B0-1B7	1B0-1B7
	4	1B8-1BE	1B8-1BE

[a]*2BF is the low interrupt vector register.*
[b]*1BF is the interrupt vector register.*

The I/O addresses for ports 3 and 4 do not change when the FourPort/XN is reconfigured for compatibility mode; DOS does not recognize ports beyond COM2.

TABLE 4: *Quadram Quadport-AT I/O Addresses*

I/O ADDRESS MAP 1 Port	Addresses	I/O ADDRESS MAP 2 Port	Addresses
A	280-287	F	288-28F
B	290-297	G	298-29F
C	2A0-2A7	H	2A8-2AF
D	2B0-2B7	I	2B8-2BF
E	2C0-2C7	K	2C8-2CF

Described above is non-DOS-compatible mode; in compatible mode, ports A and B emulate COM1 and COM2, and the remaining ports cannot be accessed.

TABLE 5: *Sigmation SMF/AT210 I/O Addresses*

POSITION	DESIGNATION	ADDRESSES
Main Board	COM1	3F8-3FF
(Channels 1-2)	COM2	2F8-2FF
	COM3	3E8-3EF
	COM4	2E8-2EF
Expansion Board		320-327
(Channels 3-6)		328-32F
		330-337
		338-33F
		220-227
		228-22F
		230-237
		238-23F

The address ranges for the Sigmation board are set into a PAL; the range in force is selected by board switches. Each port may be assigned its own interrupt.

TABLE 6: *Star Gate OC8000 I/0 Addresses*

PORT	STANDARD MAP	SECONDARY MAP
1	3F8-3FF	180-187
2	2F8-2FF	188-18F
3	280-287	190-197
4	288-28F	198-19F
5	290-297	1A0-1A7
6	298-29F	1A8-1AF
7	2A0-2A7	1B0-1B7
8	2A8-2AF	1B8-1BF

Switching between the OC8000's two I/O address maps is done with board switches. The ranges are set in a PAL and may not be altered without a new PAL.

first board (for a contiguous block of 64), making the process of finding a base address for the boards even more difficult. COMMBIOS supports a maximum of two Comm +4 boards. No XENIX support is provided.

A common interrupt request line is used by all UARTs. An open-collector, active, high pull-up arrangement permits boards to be stacked within the limitations noted earlier. The boards may be set to interrupt on any one of the hardware interrupt lines IRQ2 through IRQ5. Jumpers make the selection. If the base address is set to 2E0H and IRQ3, the fourth serial port can be recognized by the system as COM2.

The software documentation provided with the COMMBIOS software (by Commtech, Inc.) is complete and well written. The level of detail and coverage is designed for users who will set up and operate the software in a variety of situations. Interfaces to Pascal, C, BASIC, and FORTRAN are provided, along with many example source and object files and some executable programs.

The original hardware documentation from Sealevel is sketchy, consisting of a few pages of text and a wiring diagram with call-outs for jumper settings.

Sigma Information Systems, Sigmation Division. Sigmation's SMF/AT210 is one of the most intriguing of the boards presented in this survey. The hardware is a multifunction board with up to two serial ports on the main board and four more on an optional expander board. NS16450 UARTs are used. In addition, the board has a parallel port and up to 4MB of add-on memory. It is an AT-only board that occupies slightly more than a full slot and requires two slots when fully loaded. An unshielded ribbon cable runs from each port on the board to a male DB9 connector.

The most interesting aspect of the board's design is that each serial port can be assigned its own independent interrupt. The UARTs on the main board can be assigned to any interrupt from IRQ2 to IRQ5, and the expansion UARTs can choose from IRQ9-12 and IRQ15. Revision C of the board uses switches for interrupt selection. Earlier versions used jumpers.

I/O addresses are selected from predefined tables set in programmable array logic. Custom address tables can be obtained from the vendor to meet special needs. The I/O address allocations that are stored in the standard tables are shown in table 5.

Sigmation intends this to be an OEM-only product for bundling with custom multiuser operating system implementations. No device drivers are provided for DOS or any other operating system. The product suffers from a complete lack of software support, and Sigmation does not have plans to provide any at this time.

Only about six pages in the manual are devoted to the asynchronous ports; this material is sparse but adequate. Sufficient information is provided to permit someone who is knowledgeable about operating system interfaces to write a device driver.

Star Gate Technologies, Inc. Star Gate makes two versions of its serial port boards: the OC8000 and the OC4000. The OC8000 provides four, six, or eight serial ports using 8250 UARTs. The four- and six-port versions of the board may be upgraded in the field to the full eight-port level by adding UARTs and replacing a few integrated circuits. The interface to serial lines is via a shielded, flat cable and box with eight DB-25 female connectors. The first of the connectors has all standard RS-232 data, signal, and ground leads while the rest carry only the data and ground (pins 1, 2, 3, and 7), limiting their usefulness. In configurations using dedicated lines for terminals and data collection devices, this is usually not a problem but it does prevent full use of autoanswer modem equipment on dial-up lines.

The OC4000 provides all RS-232 lines for each port, but reduces the maximum number of ports supported per board from eight to four.

Interrupt assignment and address decoding logic for both boards are stored in socketed chips so customized versions can be installed easily. Two interrupt-mapping modes may be selected. The first funnels all interrupts through a single interrupt request line in the range of IRQ2-7. The second mode sets the board up in DOS-compatible fashion such that the first two ports look like COM1 and COM2. The rest of the ports can interrupt together on a third line (IRQ2 or IRQ5-7).

The I/O address logic is in a PAL. Two separate address maps (standard and secondary) are stored and may be selected by board switches (see table 6). Contiguous blocks of addresses are not necessary. Therefore, a UART's I/O addresses can be wedged into a space

in a machine's address map that would not be acceptable to those boards that require contiguous allocations of 32 or 64 addresses. Special PALs are available for IBM XENIX (System III), and PALs for other operating systems can be created.

The documentation is complete and accurate, although a couple of readings were required to set the board up for different combinations of hardware and software. The extensive software that accompanies these products is aimed primarily at DOS users and BASIC programmers. The OC8000 DOS device driver (written by Advanced Engineering Products) is documented separately from the hardware. It permits access to all system serial ports from DOS and BASIC as COM1 through COM8. Many other useful programs and source code samples are provided in Star Gate's documentation.

MEETING THE NEED

Each of the reviewed boards lives up to the promise of providing much-needed serial ports to the PC, XT, and AT. The AST FourPort/XN and Quadram Quadport-AT were designed for the AT, but physically fit in a PC and operate correctly with minor limitations. Only the Sigmation SMF/AT210 is physically limited to AT use because of its size and the use of the AT-only edge-connector pins in the I/O channel. If the host machine is an AT (or a compatible), the SMF/AT210 offers the use of individual interrupts for each of six serial ports, the addition of up to 4MB of expansion memory, and a parallel port.

The I/O addresses used by some of the boards (AST's and Quadram's) are only minimally configurable and may conflict with assignments of some other add-in hardware in a system. Some, particularly Sealevel System's Comm +4, require large blocks of contiguous I/O addresses. This complicates the process of finding a place in the PC's address space that does not conflict with other hardware's I/O address assignments. The use of PALs to customize the I/O address assignments on a port-by-port basis (as in the Star Gate OC4000 and OC8000) alleviates the contiguous address space problem but requires vendor support and, most likely, additional cost to the user. Many vendors may be unwilling to configure custom PALs in less than OEM quantities.

Boards that permit I/O address and interrupt level selection by jumper or switch (HOSTESS 8 and the Sealevel Comm +4) are the most convenient to install, with the reminder that the Comm +4 requires a large contiguous block of I/O addresses.

A multiport board, such as the HOSTESS 8 and Sealevel Comm +4, produced on a short card might fit next to one of the internal hard-disk cards, even in an XT or AT where the board-to-board spacing is tight. Most hard-disk cards are designed with the drive mechanism at the rear of the board. Although the drive mechanism obstructs a portion of the adjacent slot, enough room usually exists to install a short board with a maximum length of 5½ inches or less. This could be particularly useful in a PC Portable or Compaq Portable where the number of available slots for adapters is very limited.

Finally, software support is very important. AST's FourPort/XN, AMI's LAMB, and DigiBoard's DigiCom/8 are the only boards reviewed with readily available support for both IBM and SCO XENIX. Vendors of custom software may prefer to provide their own port drivers, but users who wish to install off-the-shelf XENIX implementations should make sure a supported driver exists for a multiport board under consideration.

Unless the user is prepared to deal with the intricacies of writing and installing device drivers, it is best to select products that provide such device driver support or those for which the support is provided by a reputable third party. For XENIX, either IBM or Santa Cruz Operation can be considered acceptable companies. Without the needed system-level software (device drivers), and usually some application-level software as well, the additional serial ports are inaccessible.

Expanding the number of PC serial ports is becoming more of a need than a want, and good software is the key to obtaining the best possible performance from the hardware. Specialized applications, realtime data collection in particular, can place strict performance requirements on the I/O system. Finely tuned software is needed to avoid data loss, maximize data transmission rate, and ensure data integrity.

With these points in mind, the best board among those reviewed is AST's FourPort/XN, a quality product from an established vendor with a solid reputation. Its documentation is everything it should be, and support for IBM XENIX (System III) and SCO XENIX System V is provided. It lacks DOS support for serial devices beyond COM2, but as few DOS applications demand such ports, this cannot be considered a crippling limitation.

Augie Hansen, formerly on the technical staff of AT&T Information Systems, now owns Omniware, a software development and training company.

LAMB (Local Area Multiuser Board): $595
American Micronics, Inc. (AMI)
17811 Skypark Circle #H
Irvine, CA 92714
714/261-2428
CIRCLE 344 ON READER SERVICE CARD

FourPort/XN: $495
AST Research, Inc.
2121 Alton Avenue
Irvine, CA 92714
714/863-1333
CIRCLE 345 ON READER SERVICE CARD

HOSTESS 8 Multiport Network Adapter: $695
Control Systems, Inc.
2675 Patton Road
St. Paul, MN 55113
612/631-7800
CIRCLE 346 ON READER SERVICE CARD

DigiCom/8: $679
Digiboard Inc.
6751 Oxford Street
St. Louis Park, MN 55426
612/922-8055
CIRCLE 347 ON READER SERVICE CARD

QuadPort-AT: $250
Quadram Corporation
One Quad Way
Norcross, GA 30093-2919
404/923-6666
CIRCLE 348 ON READER SERVICE CARD

Comm +4: $349
Sealevel Systems, Inc.
P.O. Box 1808
Easley, SC 29641
803/855-1581
CIRCLE 349 ON READER SERVICE CARD

SMF/AT210 Multifunction Board: $497
Sigma Information Systems
Sigmation Division
1110 N. Miller Street
Anaheim, CA 92806
714/632-0474
CIRCLE 350 ON READER SERVICE CARD

OC4000: $420
OC8000: $520
Star Gate Technologies, Inc.
33800 Curtis Blvd, Suite 109
Eastlake, OH 44094
216/951-5922
CIRCLE 351 ON READER SERVICE CARD

THE PROGRAMMER'S SHOP

provides complete information, advice, guarantees and every product for Microcomputer Programming.

GREENLEAF BARGAINS ORDER TODAY

Order before 9/31/86 and mention this ad for the special prices to the right.

Greenleaf Functions - 200+ functions in C for printer, formatting. With source. Covers all bases including screen, graphics, & DOS interface. Good doc. Specify compiler.
Greenleaf Comm Library - Full ASYNCH support with source. Xmodem, CRC, polling. Specify compiler.

	LIST	Before 9/30	After 9/30
Greenleaf Functions	185	119	139
Greenleaf Communications	185	119	139

Editors for Programming

BRIEF Programmer's Editor - undo, windows, reconfigure	PC	Call
C Screen with source	80/86	$ 75
EMACS by UniPress - powerful	MS	$299
Epsilon - like EMACS, Full C-like language for Macros	PC	$169
FirsTime by Spruce - Improve productivity. Syntax directed for Turbo ($69), Pascal ($229), or C ($239)		
Kedit - like XEDIT	PC	$109
Lattice Screen Editor-multiwindow, multitasking Amiga $100	MS	$109
PMATE - power, multitask	80/86	$149
XTC - multitasking	PC	$ 85

Atari ST & Amiga

We carry full lines of Manx, Lattice, & Metacomco.

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. All formats available. UPS surface shipping add $3/item.

Multilanguage Support

BTRIEVE ISAM	MS	$199
BTRIEVE/N - multiuser	MS	$469
CODESIFTER - Profiler	MS	$109
Halo Graphics - 115+ devices. Animation, engineering, business. Any MS language, Lattice, C86	PC	$229
PANEL - Create screen with editor, generates code. Xenix $539	MS	$239
Pfinish Performance Analyzer	PC	$249
PLINK 86 - program-independent	MS	$249
PLINK-86 Plus - incremental	MS	$369
PolyLibrarian by Polytron	MS	$ 85
Rtrieve - Btrieve front end	MS	$ 79
Screen Sculptor - slick, thorough	PC	$ 99
ZAP Communications - VT 100, TEK 4010 emulation, full xfer.	PC	$ 95

Xenix-86 & Supporting

Basic - by Microsoft	$ 279
Cobol - by Microsoft	$ 795
Fortran - by Microsoft	$ 399
Xenix Complete Development System	$1149

RECENT DISCOVERY

Lattice RPG II Compiler - Run RPG II programs developed for the System III or System 32/34/36 with little or no change in source code. Screen gen, ISAM, & direct files. No royalties. PC $639

Fortran & Supporting

Forlib + by Alpha - graph, comm.	$ 59
MACFortran by Microsoft	$ 229
MS Fortran - link to C	$ 219
No Limit - Fortran Scientific	$ 119
PolyFortran - xref, pp, screen	$ 149
Prospero - '66, reentrant	$ 349
RM Fortran - enhanced "IBM Ftn"	$ 395
Scientific Subroutines - Matrix	$ 149
Statistician by Alpha	$ 269
Strings and Things - registers, shell	$ 59

Debuggers

Advanced Trace-86 by Morgan Modify code on fly.	PC	$ 149
Codesmith - visual, modify and rewrite Assembler	PC	$ 109
CSPRITE - data structures	PC	$ 139
DSD87 - windowing, 8087	MS	$ 95
Periscope I - own 16K	PC	$ 269
Periscope II - symoblic. "Reset Box." 2 Screen	PC	$ 129
Pfix-86 Plus Symbolic Debugger by Phoenix - windows	PC	$ 249
Software Source by Atron- Lattice, MS C, Pascal, Windows single step, 2 screen, log file.	MS	$ 115
w/Breakswitch		$ 199

FEATURE

dBASE Programmers: Translate to C with Less Effort. Use dBx Translator

Users say dBx:
— "allowed me to concentrate on learning C." -new C programmer
— "takes grunt work out of conversion." -experienced C/dBASE programmer

If you need the portability, speed, and control of C, but all your code is written in dBASE, dBx is for you. Including a translator, C libraries and utilities, dBx produces a clean, maintainable translation of most of your programs. You complete translation then rewrite sections to take advantage of C power, flexibility.

Includes BTree, works with dB-C, CTree, C-Index, Phact, others. Supports C86, Microsoft, Desmet, and Lattice C. No royalties. Library source Unix portable, available separately.

Call our "translator specialist" for details. **PCDOS $350**

FEATURE

PROGRAMMING TEAMS: Manage and Control Source Versions Efficiently with POLYTRON Version Control System (PVCS)

Duplicated efforts and time wasted rebuilding and cataloging source can be avoided. Reliably delegate control for source & documentation to PVCS.

Save confusion and disk space by letting PVCS maintain the complete current version plus all of the increments, decrements, and related notes needed to rebuild any version. Maintains a complete history of changes. No experience with such a system is assumed. The documentation and examples will help you learn quickly. Privilege levels control access so the administrator knows who is working on each module. Ask about discounts for LAN and multi-keyboard access.

PCDOS $329

POLYTRON

FEATURE

EXPERT SYSTEM DEVELOPMENT: Practical, Complete, and Unlimited Features Help Smoothly Build Expert Systems with EXSYS

EXSYS, Inc. has built a stable and complete toolkit by listening to users and examining what they need. One of the first Expert System Shells for the PC, EXSYS provides the features of just about all of its combined competitors, plus the documentation and examples you will need to learn in this field.

UNLIMITED FEATURES? EXSYS supports backward chaining of IF/THEN/ELSE rules, full math support, probabilities, explanations, the ability to call external programs including spreadsheets, database managers, or custom-written front ends with data passed to and from the external program, plus the ability to handle substantial applications of up to 5,000 rules. All user input is either English text, menu selection, or algebraic expression.

The systems developed can explain why information is needed and how it will be used. The ability to "change and rerun" allows expert system modelling of problems. Written entirely in C, EXSYS provides very high speed execution and efficient memory utilization.

But if these features are not enough, use the interface to Lotus, dBASE, BASIC, C, or any other .EXE or .COM file. Already in use at over 1,000 sites with many complex and powerful expert systems developed. Several expert systems have been marketed with a low-cost runtime license. Single computer use is only $349. Unlimited runtime distribution is available for an additional $539.

PCDOS $339

Call for a catalog, literature, advice and service you can trust

HOURS
8:30 AM - 8:00 PM EST.

800-421-8006

THE PROGRAMMER'S SHOP™
128-P Rockland Street, Hanover, MA 02339
Mass: 800-442-8070 or 617-826-7531

"You've got everything I've heard of, and much I haven't! . . . Normally, I expect my money to be "fan letter" enough, but you people are SUPER!"
Shel Hall
Artell Corp.

CIRCLE NO. 151 ON READER SERVICE CARD 7/86

"If you can

EVEREX Streaming Tape Family

If you can find a company with better tape achievements, buy from them:

- Over 30,000 streaming tape sub-systems, ½ height, 5¼" form factor industry standard QIC 24 recording format, QIC-36 hardware interface and largest installed customer base worldwide.
- Over 100,000 industry standard streaming tape controllers in the field.
- The first company to release QIC-02 IBM-compatible software drivers to the OEM community.
- The first company to release an internal 5¼"half-height IBM compatible tape subsystem.
- The first company to release file-by-file retrieval from a mirror-image backup.

If you can find better streaming tape software, buy it:

User Interface

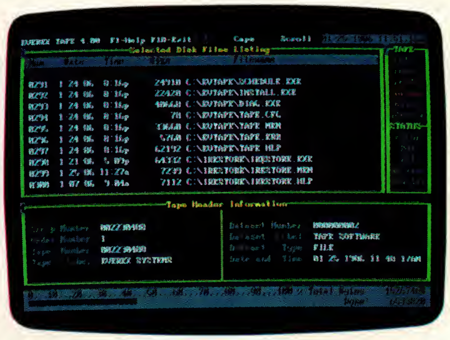
- Menu driven.
- Multiple windows (nestable and movable).
- Context sensitive on-line help.
- File selection from directory listing or command line.
- Constant tape status display.
- Menu driven hardware configuration guide.

Tape Functions

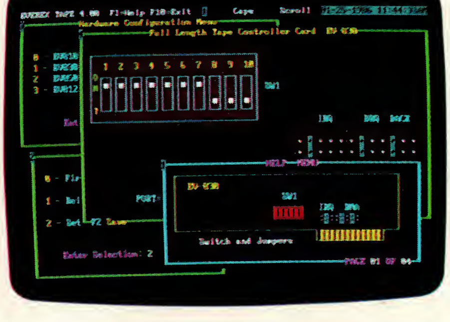
- Multiple tapes allowing unlimited backup capacity.
- Multiple datasets for multiple backups on the same tape.
- Tape sequence management recorded on the tape.
- Programmable messages.
- File-by-file and image backup on the same tape.
- Installation program.

Functionality

- Physical image backup/restore of local disks.
- Logical image backup/restore of local or remote disks.
- File backup from any local or remote disk of any file size.
- File restore from image backup or file backup.
- Allows wildcards for selection of files.
- Automatic scheduler for unattended backups.
- Backup across networks (using logical image or file-by-file).
- Real time clock display.
- Backup any disk whether formatted for DOS or not
- Escape to DOS while in backup/restore operation.



Automatic Scheduler

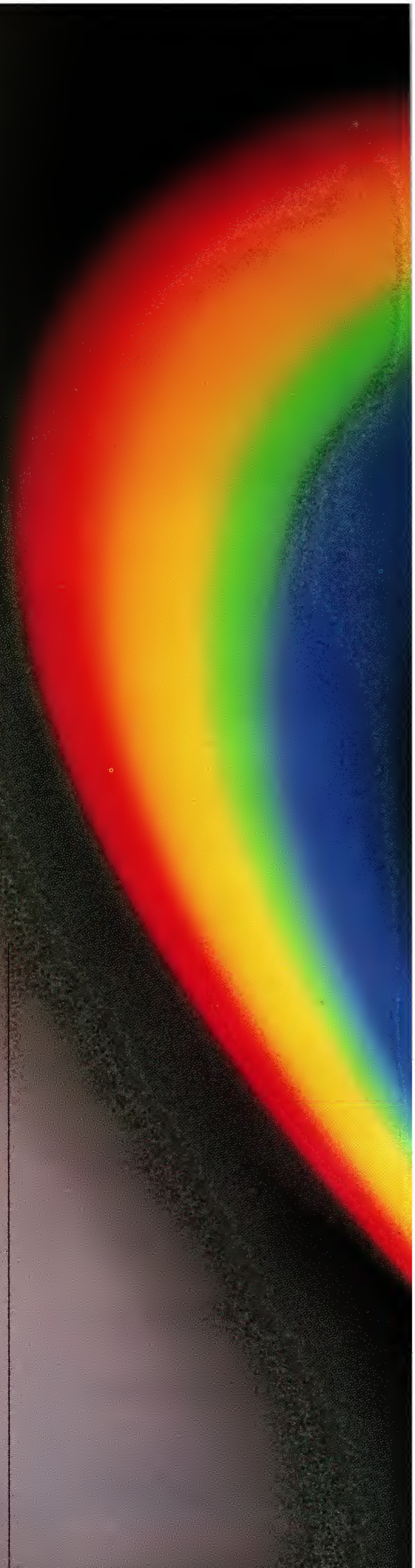


Nestable, Movable Windows

Coding in a New Light

With its elegant new language and integrated development environment, Clarion generates applications with clean design and clear code.

TED MIRECKI



The introduction of Clarion, an application generator from Barrington Systems, is noteworthy on two counts; it presents an elegant and capable language and embeds it in a comfortable development environment. The closest comparable product is Professional COBOL from Micro Focus (See "COBOL Performs," Ted Mirecki, August 1985, p.107). Clarion is more than three times the size (2.5MB) of even that large system, but at $295, its cost is less than one-tenth. Clarion is not a dialect of COBOL, but a modern language in its own right.

System requirements for Clarion are 320KB of memory, a hard disk, a parallel port, and DOS 2.0 or later. The product occupies seven disks, which are permanently write protected by being unnotched. The system is installed by a program that creates appropriate subdirectories and copies the files into them, prompting for disk changes.

The distribution disks are not copy protected, but the Clarion system as a whole is execution protected. It will run only on a machine with an actuator plugged into any parallel port. This is a pass-through device (see photo 1) that does not tie up the port for use by a printer. It is, in effect, hardware copy protection, although not the same as the hardware key variety being proposed by the Association of Data Processing Service Organizations (ADAPSO).

Using Clarion without an actuator sounds an alarm on the speaker, issues a polite message, and returns to DOS. The same protection is applied to end-user applications produced with Clarion: they will not run unless an actuator is installed. In effect, the cost of the actuator ($65 to $140 each, depending on quantity) is a licensing fee for distributing applications in Clarion.

ROOTS IN COBOL

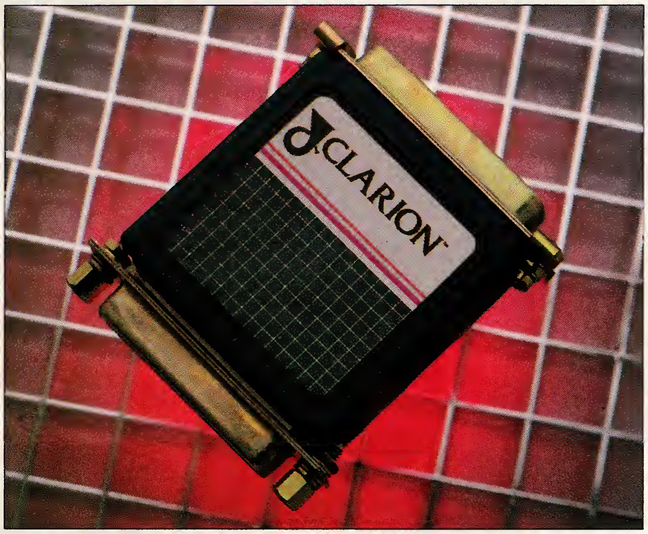
Clarion's antecedents are found largely in the COBOL-inspired features of PL/1. As a result, some of Clarion's concepts seem dated, while others follow modern objectives of structuring and declarative programming.

A Clarion program consists of two major sections: declarations and executable code. The former is analogous to COBOL's identification, environment,

PHOTOGRAPH • MARC DAVID COHEN

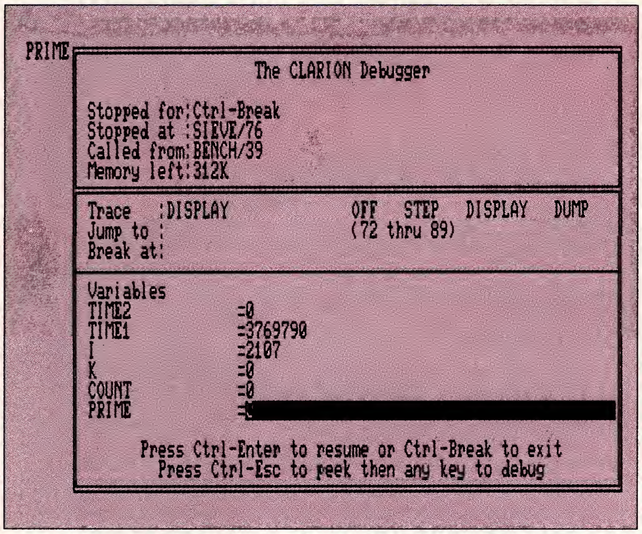

SELECT(?MEM:MOV_CODE)

PHOTO 1: *The Clarion Actuator*



The actuator attaches to a standard PC parallel port, but does not interfere with printer data passing through to the printer. It is required for running Clarion applications.

PHOTO 2: *The Clarion Debugger*



The debugger is invoked with Ctrl-Break or the STOP statement. The current screen is saved when the debugger window appears and is restored when execution is continued.

and data divisions, the latter to the procedure division. Clarion encourages the breaking up of a program into subprograms and modules (a module is a source file containing one or more subprograms). The work required of the programmer to implement this modularity, however, is a throwback to the early days of compiled languages.

Modular programs need to know the file names of various modules. In traditional languages, lists of object and library files are provided at link time, because the linker collects all the subprograms into one executable file. Clarion does not use a linker, so this information is needed at compilation time. This is provided at the beginning of the main program, in a series of statements called a MAP structure. In addition to the names of the source files, the MAP must specify all subprograms called at all levels, and which source files contain which subprograms.

A major shortcoming of Clarion's design is that it requires the programmer to supply the compiler with facts that computers can determine much more reliably. During program development, procedures and functions often are split up, combined, renamed, and moved from file to file, and the programmer must keep track of subprogram names and their locations. A reasonably smart compiler, given a list of source files, determines just from scanning the source code what the subprograms are and where they are located.

Most variables in a Clarion program are declared before being used. The data declarations in a main program define variables that are global to an entire program suite. Subprograms can also contain local data declarations.

Four numeric data types are supported: **byte**, ranging from 0 to 255; **short**, signed 16-bit integers; **long**, signed 32-bit integers; **real**, 8-byte floating point in IEEE (8087) format. These types correspond to **char**, **int**, **long**, and **double** of the C language. Conspicuous by its absence is the binary-coded decimal, or BCD, type. This could be a drawback in business applications where decimal fractions must be represented exactly, although the problem is somewhat mitigated by the **long** integer type that can represent numbers greater than 2 billion; money amounts may be expressed as integral cents, allowing a magnitude of up to $20 million. This is not enough in a large corporate setting but probably will suffice for small business applications. Any of the numeric types may be declared as arrays with up to four dimensions.

As in COBOL, strings are static. If a string is moved to a shorter destination, it is truncated; if it is moved to a longer one, it is padded on the right with spaces. The maximum length of a string is 255 characters. A string is a single object, not an array of characters as in C.

Strings also are used to hold pictures that specify how to convert numeric values into printable form, inserting editing characters such as currency symbols, commas, and slashes for dates. This is accomplished by assigning a numeric value to a string that was initialized with a picture token. These may resemble either COBOL pictures in which each picture character represents one character of the final edited value or FORTRAN or C format specifiers in which letters and numbers specify the type and length of the field. For example, either of the following strings will convert a number into four characters: @P####P or @N4.

Conversions between data types are performed automatically in assignment statements. Even a number that is formatted in a picture string may be deformatted, or converted back to numeric representation, simply by assigning the string to a numeric variable. Very few languages are capable of performing that particular trick.

Besides declared variables, Clarion allows implicit ones, which do not require declaration statements. They are defined by being used in the code section of a program, and their type is determined by the name. Variables whose names end with # are **long** integers, those ending in $ are **real**, and those with " (double quotes) at the end are strings of 32 characters. The concept of allowing both implicit and declared variables comes from PL/1, and the idea of specifying the type by the last character of the name is from BASIC.

Clarion also supports a compound data type called a *group*, which is a collection of other data types. It is analogous to a *group item* in COBOL, a *structure* in C, or a *record* in Pascal. Groups may contain other groups or arrays, and it is possible to define arrays

of groups. Several different layouts may be defined over the same area of memory, as is the case with a C union or a COBOL REDEFINES clause.

Other data structures defined in the declaration section are screen layouts, file records, and memory tables. Address pointers are not provided, but they would not be appropriate in an application language like Clarion. The functions typically implemented by pointers are automatically provided by other features of the language.

Clarion's program statements are conventional. The usual arithmetic and comparison operators are provided, as well as modulo division and string concatenation. As in C, the arithmetic operators may be applied across an assignment; for example, A += 2 means A = A+2. Group items may be copied by assignment and compared; as in COBOL, they act as strings.

A comprehensive set of structured statements is available for controlling flow of execution in Clarion, including IF...THEN...ELSEIF...ELSE, case, and five types of loops. The old-fashioned unconditional GOTO and the equivalent of COBOL's GOTO...DEPENDING ON also are available.

Another useful structured construct is the routine. This sequence of statements is executed by a DO statement and is the equivalent of the PERFORM procedure in COBOL or the GOSUB of BASIC. A routine is local to the procedure of which it is a part (meaning it cannot be performed from outside of that procedure) and shares all data variables with its enclosing procedure. The DO statement cannot pass parameters to a routine.

The syntax of the control statements resembles Modula-2 in that the language has only a block terminator (END statement), but no block initiator (BEGIN). Each control statement is itself a block initiator, as is shown by the following example:

```
IF A=B
    statement 1
    statement 2
ELSE
    statement 3
END
```

The IF...ELSE and ELSE...END pairs act as brackets for a block consisting of one or more statements.

In Clarion, the END statement is usually abbreviated to a single period. This is reminiscent of COBOL, but with a difference: in a nested control structure, a period in COBOL terminates all levels of the structure, while in Clarion a period terminates only the innermost structure. For this reason, Clarion code can be better structured than COBOL code. The syntax is more logical and cleaner than that of either Pascal or dBASE, thus making the language very easy to write and read.

Data can be passed to Clarion subprograms in one of two ways: in global variables or in parameters. Global variables are defined in the main program, as opposed to local variables, which are defined in a subprogram.

Parameters may be received either by value or reference. Those that are received by value have their types declared in the receiving procedure; if the procedure is called with arguments of different types, the runtime system performs the appropriate conversions.

Parameters received by reference are declared EXTERNAL, with no facility for specifying type or length. This is acceptable for numeric or string scalars

Clarion's syntax is more logical and cleaner than either Pascal's or dBASE's, making the language very easy to write and read.

(single items). It is not very useful for complex types such as arrays and groups because the procedure has no way to name and gain access to the component parts of such arguments. The entire argument is treated as one string. This can be circumvented by assigning the parameter at entry to a group that is local to the procedure, then passing the result back to the caller at exit by assigning the local group to the parameter. However, this procedure defeats one purpose of passing parameters by reference: to avoid the overhead of creating another copy.

Another problem with the procedure-calling mechanism is that files cannot be passed as arguments. This prevents the writing of general-purpose procedures that perform some function on any given file.

Assembly language interface is accomplished through binary modules, which are similar to .COM files ORGed at 0H rather than 0100H. Binary modules must contain headers describing the names and number of procedures in the module, along with the number of parameters passed and a descriptor for each parameter. The documentation on this topic is sufficiently clear and complete for the moderately experienced programmer.

In screen processing, Clarion may truly be considered a fourth generation language in the sense that the program code specifies the desired effects, not the details of how to achieve them. The language provides comprehensive console I/O facilities with amazing economy of a programmer's time and effort.

The layout of a screen is defined in a data structure whose elements are the individual fields: display-only fields (including box-drawing lines and corners), entry fields, menus, and pause fields that direct the user to press a key to proceed. Row and column coordinates and video attributes may be specified for all of these. An editing picture, a variable to receive the input data, and the name of a help screen also may be specified for the input fields. Coding the declarations of the screen structure can become very tedious, but a screen painting utility automates the process to a certain extent.

The beauty and power of Clarion's screen processing is that, during console I/O, complex actions occur automatically as a result of simple statements in the declaration section of the program. In other languages, the same effects are typically achieved by writing executable and usually quite complex code. For example, console I/O is initiated by opening a screen structure. If the structure is declared as a window, the OPEN procedure saves the underlying display before opening the new window. Closing the structure automatically restores the previous display if the structure is a window or clears the screen if the structure is a full screen. This process may be repeated as many times as memory permits, piling up screen windows on top of each other.

All of the cursor control operations involved in point-and-pick menu selection are automatically available merely by performing an ACCEPT procedure on a group of fields declared as a menu structure. Navigation around the menu entries may be accomplished either via the cursor keys or by typing the first letter of a menu entry. Two values are returned to the program from the menu: the number of the field selected and the text from that field.

Declarations also can be used with individual fields to specify actions to be taken when the operator presses certain keys while the cursor is within that field. The Esc declaration names a field

that is the predecessor of the current field. When the operator presses the Esc key, the cursor is then returned to the predecessor field.

Field-level editing is provided, meaning that the application program regains control from the runtime code after each field is entered. This allows varying the processing of the remainder of the screen on the basis of the user's input. The application can react immediately to erroneous input by displaying a help screen or list of valid data. Fields can be referenced either by number, which is useful for processing them in loops, or by name, which prevents maintenance problems when fields are added, deleted, or reordered.

Clarion's file processing is the antithesis of its screen processing. Whereas complex console I/O is performed with a minimum of code, file I/O requires detailed record-level coding of the COBOL type. The effort required of the programmer in coding file I/O is at odds with the otherwise modern concepts of Clarion, such as the lean syntax of the language and the declarative programming of screen processing. For example, updating in a file all the records that meet certain conditions requires writing a loop that reads each record, tests it against the condition, changes its values, and rewrites it. A typical data manager is able to accomplish the same task with one statement.

Clarion's file capabilities are by no means disappointing, however; they are powerful and comprehensive, if a little slow. In most cases, Clarion programs act on data files with a proprietary header and format. These Clarion format data files cannot be created by a running program, only by a utility that is provided with the system. File space need not be preallocated for a specific number of records, however; the utility merely sets up the headers that are required by Clarion files. The header's format is not documented.

One underlying file organization supports sequential, relative, and indexed access to data. All data files have fixed-length records, and keys (any number per file) are kept in separate files. The same data file may be accessed by any of the three methods, even within the same program. Utility programs are provided to maintain, rebuild, and sort data files. Indexes may be created along with the file or built later from the data.

Relative (also called direct) access retrieves records by number. Clarion's implementation is different than most because it allows retrieval not only of the *n*th record in the physical file sequence, but also the *n*th record in the logical sequence by any of the keys. In effect, the program can retrieve keys as well as records by number.

With indexed access, records may be retrieved by any of the keys at any time. All keys are equivalent, and any or all may allow duplicates. Any key may be changed by rewriting it; the job of moving the index record in the key file is handled automatically by the system.

Besides the data files in its proprietary format, Clarion also processes generic DOS files. These may be standard text files (the only variable-length record Clarion supports), byte stream files, or fixed-length record files. DOS files can be created, renamed, and deleted from within programs.

Clarion's file capabilities are by no means disappointing, however; they are powerful and comprehensive, if a little slow.

By default, all files are closed at the operating system level after every write, providing a high level of data integrity in interactive systems. To improve efficiency in batch processing, the program may keep the file open in stream mode.

Clarion also offers additional efficiency measures not available in other high-level languages. A buffer may be declared to hold multiple records from the file, thus reducing disk accessing in stream mode. The buffer space is not declared in the program's data area, but is allocated at runtime in the extra system memory outside of the program's initial allocation. The size of the buffer may be specified in kilobytes, in number of records, or, most useful of all, as a percentage of remaining memory space. Often-referenced keys or records also may be kept in a cache outside of the program space.

The language provides record-like processing of in-memory tables. The tables are implemented as doubly linked lists in the extra memory outside of the program's data space. The list pointers and processing are handled automatically by statements that resemble those for file processing: ADD an entry to the table, GET a table entry, PUT an entry into the table. Entries may be referenced by key or by entry number. The tables can grow to fill up all of the available memory beyond the program's declared data space. Any buffers and caches are deallocated as needed.

Another type of data structure is the report, consisting of headers, footers, and detail lines. Its purpose is only to specify the horizontal layout of the various parts of a report. It does not handle COBOL-style Ctrl-Break processing (for example, printing a subtotal when a key field changes), that must be programmed. Print line layout may be specified in string variables equally as well, but with the report structure, line counting and page overflow is automatically handled, and the structures defined as headers and footers are automatically printed. The report structure also is used by a layout utility for designing reports on screen.

DEVELOPMENT ENVIRONMENT

The development environment is implemented as 11 utilities that are usually invoked from a main menu, although some of them may be run directly from DOS or from batch files. They include the Editor; Compiler; Processor (executes compiled code); Screener (generates screen layouts); Helper (generates help screens); Reporter (generates report layouts); Filer (creates and modifies data file structures); Scanner (browses and updates data files); Sorter (sorts and rebuilds data files); Director (displays directories and executes DOS commands); and Tailor (changes Clarion's default settings).

The user interface is common to all of the utilities. Its most notable characteristic is that it is consistent with the Clarion language; any of the system's input screens and responses can be programmed in Clarion. The developer therefore knows in advance the subjective feel of the application's interface, because it is the same as the one used for development.

Each utility presents an initial screen on which the user enters file names and other appropriate information. Most information on these screens is saved on disk and is preentered when the screen is displayed; it is available from one session to the next. For example, the compiler's screen shows the last file edited, and the runtime system displays the last file compiled.

After filling in as much of the screen as is appropriate, the user may start the utility by pressing Ctrl-Enter or may back out to the menu with Ctrl-Esc. Interactive utilities such as the Editor are controlled by Ctrl-letter combina-

tions; these are meant to be mnemonic, not positional as in WordStar. With only 26 alphabetic keys, however, the connection between a function and its letter is inevitably remote in some of the more feature-laden utilities; further, the same letter performs different functions in different programs. To alleviate confusion,the first help screen in each utility lists all the Ctrl-letter combinations for that program.

Exit from a utility is with Ctrl-Enter to save whatever work the utility did or with Ctrl-Esc to abandon it. In the latter case, the user is asked for confirmation.

Seven of the 11 Clarion utilities may be invoked by pressing function keys. Doing so from within another utility avoids loading the menu program. Pressing a shifted function key bypasses the initial screen and immediately begins processing the last file processed by that utility. One function key is reserved for a compile and run sequence. Pressing this key in the editor utility saves the source file, terminates the editor, compiles the program, and runs it if the compilation is clean.

Field editing during screen input is somewhat quirky. The Enter key advances from field to field and the Esc key backs up to the previous field. These maneuvers are fairly conventional. In numeric fields, however, NumLock is automatically turned on and cannot be toggled off. This means that cursor control is unavailable in numeric fields, and only the low-order digit may be changed. A high-order digit may be changed by erasing and reentering all the digits to its right.

In character fields, the Backspace and Delete keys work as expected, but the use of the Ins key is unusual; it inserts a single space at the cursor and pushes the rest of the field over to the right. Holding it down inserts multiple spaces. The action parallels that of the Delete key, which deletes the character at the cursor and shifts the field left. Both produce repeating action, and neither moves the cursor.

Editor. Despite some minor annoyances and lack of advanced features offered by the better stand-alone products, Clarion's editor provides adequate program maintenance facilities, and after an initial learning period memorizing the sometimes arbitrary mnemonic command structure, it becomes fairly easy to operate. Users may invoke their favorite editor or word processor outside of Clarion, because the source files are standard ASCII text.

The editor uses a configuration file that defines keyboard macros and editor defaults. As in IBM's mainframe ISPF editor, default configuration files are identified by the extension of the source file, so a different configuration may automatically be used for each file type. The usefulness of this feature is compromised by the fact that the default configuration file is found only if it resides in the current directory; otherwise its full path must be entered in the editor's initial screen. The configuration file name is not saved from the last editor session.

The editor's speed of operation is not spectacular, but acceptable. Screen updates seem to be done through the ROM BIOS, because when using a color/graphics adapter, paging is slow, but free of video interference.

The most frustrating part of using the Clarion editor is waiting for all 103KB of it to load. When entering the editor to correct a minor error, more time is spent in loading than in actually making the correction. The size seems excessive, given that very few editor features are implemented.

The user interface is common to all of the utilities. Its most notable characteristic is that it is consistent with the Clarion language.

The basic features of the editor include: undelete, search and replace, cut and paste, write out to and merge from other files, automatic indentation, and column-oriented alignment. The last feature maintains the column alignment of words on a line even when the words are shortened or lengthened, which is very useful for Clarion's column-oriented syntax. Tabs within the source code are saved as tabs, not expanded to spaces, thus saving file space in columnar programs.

Cut and paste commands work on entire lines only, not on words or columns. The procedure is simple and intuitive. The cursor must be positioned at one end of the block before starting the block move or copy, but after that, prompts lead the user through the proper sequence of highlighting the block, pointing to its destination, and completing the operation. No other commands may be issued until the block operation is completed.

The editor lacks many features of more advanced stand-alone editors: it cannot search for regular expressions or wild-card strings and does not support windows. Keyboard macros are available, however. Each macro contains up to 40 characters and consists of any combination of text and editor command keys. Macros are saved in the editor configuration file and thus may be customized for each type of file on which the editor operates.

Compiler. Clarion's compiler is not a true native code compiler. Its output is an intermediate code that is interpreted by a runtime processor. Depending on the nature of the program, the size of the intermediate code may be anywhere from one-half to four times the size of the source code. In addition, the compiler produces a map file that contains global data and pointers to the called subprograms. No link step is needed between compilation and execution.

After compiling a main program file, Clarion offers to stream compile all of the files named in the program's MAP structure. Only the files referencing global data need to be recompiled after a change to the main program, but neither the MAP structure nor the response to the stream compilation query allows any distinction to be made between procedures that need recompiling and those that do not. Precompiled libraries of general-purpose procedures cannot be maintained, because a change to a main program that affects any of the called files requires the recompilation of all of them. A separate output file is generated for each source module in the compilation stream.

The handling of compile-time errors is one of Clarion's best features. Errors are not reported as they are encountered but are written to an error file. As each source file compilation is completed, a message listing the number of errors is displayed. At the end of a main program file, the user can enter the editor to correct any errors or to proceed with stream compilation or other processing. If the correction option is chosen, Clarion calls the editor, loads the file just compiled, positions it at the first error, and displays a short error message. The editor has a command to jump to the next compiler error; it will not lose track of error locations even if lines are added or deleted in the course of making corrections. Because of this integration between editor and compiler, the user seldom needs to refer to a compilation listing.

During stream compilation, an error count is displayed as each source

file is completed, but the option of correcting errors is not given until all the files in the stream have been compiled. Then, a summary of error counts for each file is displayed, and the user has the option of entering the editor to correct the first file where errors were found. After these are corrected, the list of files with errors cannot be conveniently reviewed. In general, however, Clarion's compile-time error-handling works very smoothly, greatly speeding up the edit-compile process.

Processor. The Clarion processor is the runtime system that executes the intermediate code produced by the compiler. At 243KB, it is the largest of the Clarion modules, partly because it incorporates the debugger, but mostly because it contains all of the routines that in other languages reside in libraries and are linked into each user-written program. This kind of a runtime system saves space by keeping only one copy of each routine instead of a duplicate copy in each module that needs it, but all the routines are present at least once whether or not they are needed.

One of the primary functions of a runtime system is the handling of runtime errors. In this regard, Clarion leaves much to be desired. For most file I/O errors, it merely posts an error flag to be tested by the user program. The runtime does not care whether the error is in fact processed, and takes no action if it is not; thus, a program that does not test for errors may proceed to a seemingly normal termination even though catastrophic failure may have occurred. This is not a bug, but a conscious design choice. Barrington Systems assumed that all programs would be written to check for and handle file errors, but then disproved that assumption by writing a very important one—the runtime system itself—that did not. The company has since reconsidered and is changing the runtime system to abort in case of errors that are not corrected by the user program.

Error handling is also inconsistent. An error on a call to the OPEN procedure (file not found, invalid name) is posted in the internal error flag while allowing execution to continue, but an error on an implicit open (any I/O operation to a closed file) causes termination of the program with an error message. This means a program *must* test for errors after an explicit OPEN or they are ignored, but *cannot* test for errors after an implicit OPEN. Failures, such as device not ready and read or write errors, are handled automatically by the runtime system. The user is given the choices "Abort or Retry" (but not ignore); the program cannot intercept them to provide other alternatives. An ideal user program should be able to process any error, and the runtime system should apply default processing only to those ignored by the program.

The Clarion processor debugger is activated either by pressing Ctrl-Break while a program is running or by the execution of a STOP statement in the program. The debugger's screen, shown in photo 2, allows setting the TRACE

Helper's ability to select the correct symbol almost all of the time borders on the uncanny. This is the best of Clarion's utilities.

mode, resuming execution at a given point in the program, setting as many as eight breakpoints, and specifying up to six variables whose values are to be displayed. The debugging screen overlays the program's output screen, but does not destroy it; the two screens may be swapped at will. Other than the ability to refer to variables by name, the debugger has no symbolic or source-oriented capabilities. All references to the source program are by line number, meaning that a current compilation listing is required for debugging.

The TRACE mode output is similar to that produced by the TRON command in BASIC, but instead of showing all of the statement numbers in the path of execution, it displays only the jumps—a call to or return from a subprogram, GOTO, LOOP, IF, CASE, and so on. The step mode of TRACE stops the program and redisplays the debugging screen at the next jump. The display mode stops every 76 jumps, displaying a screen of statement numbers where those jumps occurred. Unfortunately, this display cannot be recalled once it is replaced by the debugging screen or the program's output screen. The dump mode runs continuously, writing the record of jump points to a disk file that then may be printed or displayed with an editor.

Executing programs in debugging mode exacts a significant time penalty. With TRACE on, the debugger took 145 seconds to run 76 iterations of the first loop in the SIEVE procedure (see listing 1, BENCH.CLA) while furiously accessing the disk.

This debugger is quite primitive when compared to symbolic debuggers supplied with some COBOL or other language compilers, but it is better than the debugging tools with other interpreted systems, such as dBASE.

Helper. This special-purpose editor creates and modifies display-only text screens. Besides the text itself, Helper records information about the size, location, and video attributes of the help screens. Helper is able to store any number of individual screens within a single help file.

The sign-on screen presents fields for the file name and screen name. Other information on Helper's sign-on screen is the location of the help window on screen; if this is entered as "floating," Clarion places the help window (provided it is small enough) in a location that will not obscure the field that contained the cursor when the Help key was pressed. Clarion's runtime system handles this relocating with no effort on the part of the programmer.

Another useful Helper feature is the ability to chain screens together to provide multipage help. At runtime, when a screen that is part of a chain is displayed, pressing Enter displays the next screen in the chain. Regrettably, the user cannot back up through the chain to display the previous screen.

The process of laying out the help text is straightforward. The cursor control keys work as expected; text may be directly typed anywhere on screen; and any rectangular area may be moved to another location. Each window is assigned an overall attribute for default foreground and background colors, but any part of the window may be given other attributes by painting with the cursor. The user does not have to specify separate screens for color and monochrome systems. Color attributes are ignored and a default set of black and white attributes is automatically substituted if a monochrome adapter or black and white mode on a color/graphics adapter is detected at runtime. Single and double line boxes may be drawn with a minimum of keystrokes; appropriate corners and intersections are supplied automatically. Helper's ability to select the correct symbol almost all of the time borders on the uncanny. Overall, Helper is the best implemented of Clarion's 11 utilities.

Screener. Clarion's Screener is used to lay out data input screens for application programs. The name of a program and of a screen structure within it must

be specified on Screener's sign-on screen. If the screen name is left blank, the user is presented with a list of screens defined in the program and may choose one for processing by moving the highlight to it with the cursor control keys. Entering a screen name that is not defined creates a new screen structure within the program. Screener writes its output back to the source file in the form of a completed screen structure with field labels and their row-column coordinates.

Sizing and locating the window, establishing video attributes, entering constant text, and drawing lines and boxes work as they do in Helper. Defining the variable fields and menus is somewhat less convenient. Instead of indicating the location of variable or menu fields directly on the screen, the user describes their locations and labels by making entries in a table. The format of these entries is very similar, but not identical, to the syntax of screen definition statements in the source language. The advantage of using the Screener utility instead of coding the screen structure directly in the source program is that Screener immediately displays the resulting layout, with dashes in the place of variable fields, and allows rapid cycling between viewing the screen and editing the table of field definitions. An obvious improvement would be to dispense with the table of nonstandard field definitions and to allow direct editing of the source statements that Screener generates.

This screen generation utility gets the job done but is not commensurate with Clarion's powerful and convenient runtime screen I/O capabilities.

Reporter. In the same way that Screener creates I/O screen structures, Reporter creates report structures. Whereas titles and other constant information are entered directly, variable field information for the headers, footers, and detail lines is entered in tables, in a format slightly different from that used in the source language. The layout may be quickly fine-tuned by alternating between editing the table and viewing a screen image of the resulting layout. Horizontal scrolling supports reports wider than the screen. Lines and boxes may be drawn on the report, if the printer supports the IBM graphics characters.

Filer. The need for Filer comes from the way Clarion keeps track of the layout of data files. In most traditional programming languages, file layouts are defined within the program. Data managers such as dBASE record file layouts in the data files. In Clarion, however, they must be present in both places. The first function of the Filer utility is to create empty data and key files from the file structures in Clarion programs. One problem with this method is that a program cannot create a new file at runtime, but all file I/O must be performed to existing (possibly empty) files created with Filer. This could be inconvenient in cases where a program needs to create a new data file, such as during a monthly or annual close in an accounting application.

The duplicated file layout information creates the possibility that the file structure as recorded in the file is different from that defined in the program. This causes a runtime error. A second function of Filer is to modify existing data files to conform to any changes to the file's structure defined in a program. Such changes may involve adding or dropping fields, renaming fields, and changing field types or lengths. This function is the equivalent of the MODIFY STRUCTURE command contained in dBASE.

Scanner. Clarion files in tabular format similar to a spreadsheet are displayed by the Scanner utility; it is analogous to the dBASE BROWSE command and just as useful for ad hoc file maintenance. Unlike dBASE, however, Scanner does not limit the width of each displayed field to the width that is defined in the file. The widths, titles, and order of fields may be changed for display purposes without changing the file structure, and this display format may be saved in a separate file so that it does not need to be reentered every time a data file is scanned. In columns that are narrower than the field, the contents may be scrolled horizontally.

A Clarion data file may be displayed in the physical record order, or, for an indexed file, in the logical order of any of the keys.

A Clarion data file may be displayed in physical record order, or, for an indexed file, in the logical order of any of the keys. Records may be added, changed, marked for deletion or unmarked. As in a data manager, each change to the file is immediately written out to disk, so the user has no choice of saving or discarding changes, as with a spreadsheet. Navigation through the file is by cursor keys, record number, or a search for field contents. The SEARCH function is somewhat limited when compared to that of an editor, because only the leading characters in a field can be found. Scanner is more capable (although slower) than the SEEK function in a dBASE BROWSE because it can search for fields other than keys, and it can replace whatever it finds.

One particularly odd characteristic of the Scanner display is that it is permanently in Scroll Lock mode. The record at the center of the screen is highlighted, and the cursor keys scroll the records through the stationary highlight instead of moving the highlight through the records. This mode cannot be turned off.

Scanner also can display and modify DOS files. With these, its operation is similar to that of BASIC's DEBUG—a hex display in the left two-thirds of the screen and an ASCII text in the right one-third. One advantage over DEBUG is that file contents may be changed by moving the cursor to the desired location in the file and overtyping in either the hex or ASCII sections.

Sorter. The primary purpose of this utility is to reorganize Clarion data files by removing deleted records, sorting into a specified physical sequence, and rebuilding index files. The sort order may be specified either in terms of field names or record offsets, although the value of the latter is questionable. The number of sort keys is unlimited, and either ascending or descending sequence may be specified for any key. Unlike stand-alone or COBOL internal sorts, Sorter cannot perform record selection or merging; it can concatenate several data files with the same record layout, with or without sorting.

Compressed files are useful mainly for backup or transmission to another system because they must be expanded by Sorter before they can be used by a Clarion program or utility.

Director. Clarion includes a DOS shell that displays a sorted directory list of any specified subdirectory. The sort order may be by name/extension, extension/name, size, date, or time, and the display may be limited to the files that match a specified wild-card pattern. Multiple files may be marked for copying, deleting, renaming, typing (to the screen), or printing. A compiled Clarion program file may be selected for execution by the processor. This selection capability is not provided for the other

DAN BRICKLIN'S DEMO PROGRAM HAS EVERYONE TALKING.

Dan Bricklin's Demo Program is a new concept in prototyping and demo-making. With DEMO you can conceptualize programs, describe them to others, refine their functionality and human interface, and finally, teach users the finished product. The two most common applications are to prototype a product before it is written and to produce a demonstration or tutorial about an existing product or system.

- Create, edit and view a series of slides
- Each slide is 80x25 characters
- All 256 characters supported, including special characters
- All 256 attributes (color, underline, blink, etc.)
- Each slide can be the image of what a running program looks like
- Characters graphics only, no bit-mapped graphics
- Slides are stored compacted, allowing often hundreds and hundreds of slides to fit in memory for instant access (given enough RAM)
- Complete set of editing commands, including cut, paste and move
- Line, box and special character commands built-in
- Includes background "CAPTURE" program for importing screens from running programs
- Learn-mode macro facility
- Run command switches from slide to slide under automatic control
  Options include:
  - ☐ Adjustable delay between each slide
  - ☐ Wait for key or optional timeout
  - ☐ Display different slide next depending upon keystroke
  - ☐ Produce tones
  - ☐ Switch to another set of slides in another file
- Runtime-only version of program included, along with license to make up to 50 copies (see License for details).
- Additional volume runtime licenses available.
- Requires 256k IBM PC/compatible, DOS 2.0 or later. Supports Monochrome, Color/graphics, and EGA Adaptors (text mode only).

Thousands of developers are designing better products faster and producing more effective demonstrations using Dan Bricklin's Demo Program. You can, too. Act now!

ORDER YOUR COPY TODAY

ONLY $74.95

617-332-2240

Send checks to Software Garden, Inc., Dept. T, P.O. Box 373, Newton Highlands, MA 02161. Massachusetts residents add $3.75. Outside the U.S.A. add $15.00.

"A winner right out of the starting gate... After you use DEMO once, you'll wonder how you got along without it."—PC Magazine, 4/29/8

PRODUCT OF THE MONTH
—PC Tech Journal, 3/86

"Everybody who writes software, either commercially or for in-house applications, should immediately order a copy. Period. No exceptions."
—Soft•letter, 4/20/86

"Apparently has a hit on its hands with... a development tool for personal computer software that has won rave reviews from early users."
—Computerworld, 4/7/86

"Its low price, superb performance, and range o applications practically guarantee that it will be widely used. Four Floppy Rating (8.0)"
—InfoWorld, 3/31/86

SOFTWARE GARDEN

"A gem."
—PC Week, 3/18/86

utilities where it would be especially useful to select from a directory list a source file for editing. As is the case in Scanner, the display is permanently in the Scroll Lock mode.

In addition to the key-driven operations on the directory list and on marked files, Director provides a command line that accepts any DOS command with one exception: a second copy of Clarion cannot be started. The memory available to DOS is reduced by more than 157KB. This seems excessive but is in keeping with the bloated sizes of all the Clarion components. The command line input may be edited with nondestructive cursor moves, insert and delete—a significant improvement over the limited editing capabilities available at the DOS prompt.

Unlike most DOS shells, Director has been designed for the experienced DOS user. It adds features and convenience, yet maintains easy access to the full power of the operating system.

Tailor. This configuration utility sets default parameters used by Clarion. With Tailor the user can set default extensions for all the Clarion file types, source code format for the statements generated by the Screener and Reporter utilities, automatic compilation for source listings, and screen colors to be displayed by the Clarion utilities. As in user-written programs, screen colors are overridden if Clarion is run in monochrome or black and white mode.

THE APPLICATION ENVIRONMENT

Clarion programs are semicompiled to pseudocode and need to be executed by a runtime system. The minimum system consists of the Processor, but the end user can put the functions performed by the Scanner and Sorter utilities to good use. In fact, Barrington Systems encourages developers to install the entire Clarion system at each site that runs Clarion applications in order to ease on-site program maintenance. The royalty for installing all or part of the system is the same: each machine running any part of Clarion must have an actuator plugged into a parallel port.

As a runtime system for end users, the processor has one drawback: the Break key invokes the debugger. Although the debugger's screen plainly states how to resume normal execution of the interrupted program, the remote possibility exists that an inexperienced user experimenting with the debugger might accidentally destroy file data. Clarion should offer a version of the processor without the debugger, strictly for executing applications.

Apart from this idiosyncracy, Clarion can produce some very user-friendly applications. Because Clarion programs can execute any DOS command, its powerful and convenient screen interface can be used to create a DOS shell for performing operating system functions or for running other applications. The overhead is only 24KB—much less than most commercially available shells. The price to be paid is that the processor's context is written out to disk before running DOS and reloaded afterward, resulting in a delay. Performing this swapping to a RAM disk would speed up the program considerably, but at this time, Clarion will write the context only to the current directory on the default drive.

Clarion's powerful and convenient screen interface may be used to create a DOS shell for performing operating system functions or for running applications.

Designing the console interface is an art in application development. The developer using Clarion has a ready model, the same interface used in the development environment. A minimal programming effort provides the user with all screen and keyboard facilities available when the program was created. The commonality between the development and applications environments is Clarion's major triumph.

MIXED PERFORMANCE

The program used to test Clarion's speed in arithmetic and file operations is shown in listings 1 and 2. The file I/O procedures are translated from an earlier COBOL version (see "COBOL Performs," June 1985, p. 79). The split between the two source files is arbitrary; it serves to demonstrate the MAP structure for programs consisting of more than one file.

Benchmark results are listed in table 1. The tests were run on two computers: an IBM PC with 640KB and a 20MB hard-disk drive, and a 6-MHz IBM PC/AT with 640KB RAM and a 20MB hard-disk drive.

Although Clarion will not typically be used for compute-intensive applications, its literature does suggest it as an alternative to FORTRAN, BASIC, and Pascal. Therefore, the results of sieve and real arithmetic tests are compared to those obtained with BASICA. In file I/O, Clarion is most like COBOL, so its performance is compared to Ryan-McFarland COBOL, another semi-compiled language. For comparison to other COBOL compilers, see "COBOL Performs," August 1985, p.107.

Clarion's performance results are mixed. Whereas the Sieve test takes almost twice as long as it does in BASIC, the arithmetic test is almost twice as fast. In file I/O, the results are roughly comparable to RMCOBOL's.

The benchmark programs attempt to optimize Clarion's file I/O by the use of stream, buffer, and cache procedures provided in the language. Without these, the times were much longer, in some cases by a factor of 10. Clarion's default file I/O support performs a DOS close and reopen on a file after *each* write to the file so that a system crash will destroy no more than one record. Clarion's file I/O optimizations are most useful in batch processing where large numbers of records must be handled in rapid sequence; in the interactive mode that Clarion is best suited for, the default I/O procedures are much safer and do not noticeably degrade response time.

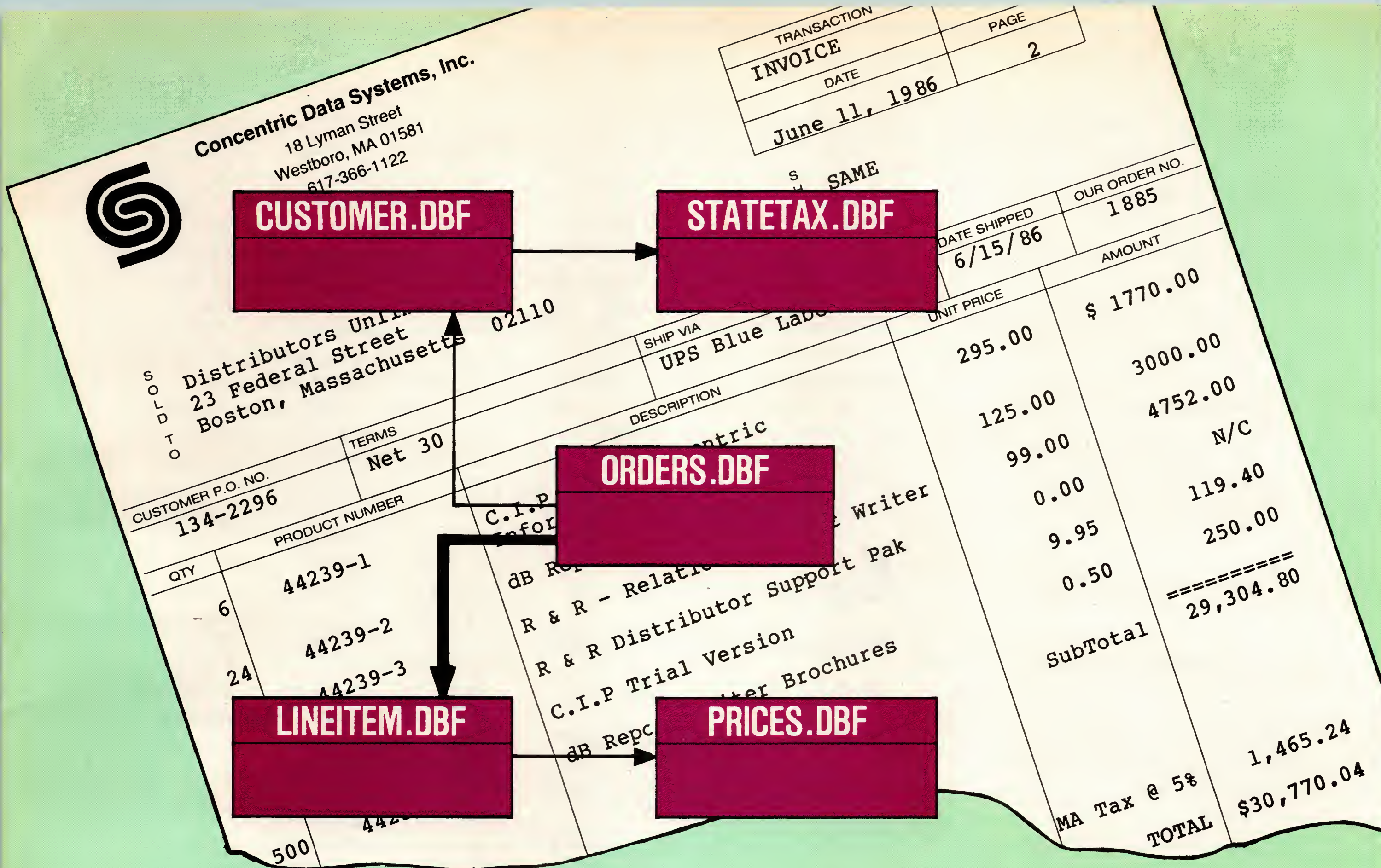
In batch mode, even with the optimizations applied, Clarion is no speed demon on large files. Batch processes for Clarion files should be written in other, faster languages, but unfortunately Clarion's proprietary data file format is not documented.

Clarion's three volumes of documentation include a reference manual that defines the language, an operating guide that describes the development environment, and a tutorial.

The language manual is the least effective; it is strictly a reference tool, containing no explanations whatsoever. Clarion is a large language and introduces some novel concepts that need discussion and examples beyond mere statements of the syntax. It cannot be adequately described in 150 pages.

The operating guide is a little better, giving fairly complete instructions on how to operate the various components of the development system. Still, it has problems; finding out how to do even simple tasks, such as adding an item to a menu, can be difficult. The index is not very detailed.

Of the three volumes in Clarion's documentation, the tutorial is the most useful. It presents 13 example programs



Relate and Report. Then Rest and Relax. With R&R, the Relational Report Writer™ for dBASE.®

The report writer dBASE is missing.

R&R helps prepare reports faster, easier, without programming — and without dBASE. Or print reports from your own dBASE programs with the R&R Runtime, *included with an unlimited-use license.*

More flexibility than you'll ever need.

Relate and report from up to 10 files at once, using one-to-one relations and one-to-many scans through dBASE indexes.

Free-form layout of text and fields anywhere on the report.

Calculate new fields with more than 60 predefined functions, including IF/THEN logic.

Eight sort levels. Eight independent record grouping levels. Compute running totals, or subtotals by group or by page.

Plain-English query, even allows date "wildcards".

The ease of use that saves you time.

1-2-3®-like commands. Move fields and text freely around on the screen. Timesavers like automatic trim and alignment.

Predefined field formats such as currency and commas, or create your own. Word-wrap character and memo fields within any width you specify.

Print selectively: underline, *italic,* **bold, combinations.**

Saved reports work even after file STRUCTUREs are MODIFYed.

Instant on-line HELP. Fully-indexed User's Guide. Tutorial.

Only $99. Satisfaction guaranteed.

Try R&R. If you aren't satisfied, just return it within 30 days for a full refund.

Order now. Call 800-325-9035.

Or call (617) 366-1122. Visa, MasterCard, check, COD, PO. Add $3 shipping, plus $2 if COD and 5% tax in Mass. Refunds only for purchases made directly from Concentric and Concentric Authorized Dealers. Works with dBASE III® and III PLUS.™ Not copy-protected.

Relate and report, then rest and relax — with R&R. A timesaving tool from the authors of 1-2-3 Report Writer,™ dB Report Writer,™ and C.I.P® File Manager.

R&R Relational Report Writer™

Concentric Data Systems, Inc.
18 Lyman Street, PO Box 4063
Westboro, MA 01581-4063

Table 1: *Benchmark Results*

	PC[a]		AT[b]	
	CLARION	RMCOBOL	CLARION	RMCOBOL
SEQUENTIAL I/O				
100 Records				
Write	7	5	2	2
Read	3	4	1	1
Copy	8	16	2	3
300 Records				
Write	21	15	5	4
Read	6	13	2	3
Copy	24	27	6	7
DIRECT I/O				
100 Records				
Read[c]	7	24	2	5
Update[c]	18	53	3	10
300 Records				
Read	20	66	5	15
Update	48	131	9	32
INDEXED I/O				
100 Records				
Write[d]	177	246	32	45
Read[e]	18	48	6	15
Update[f]	208	287	39	55
300 Records				
Write	630	386	115	67
Read	66	136	24	19
Update	695	574	142	80
	CLARION	BASICA	CLARION	BASICA
ARITHMETIC				
Eratosthenes sieve	409	203	146	70
Multiply/divide	161	252	54	87

Times are in seconds.
[a] *The configuration for the test machine was 4.77-MHz 8088, 640KB of memory, no 8087 math coprocessor, file I/O to IBM floppy disks.*
[b] *The configuration for the test machine was 6-MHz 80286, 640KB of memory, no 80287 math coprocessor, file I/O to IBM 20MB hard disk.*
[c] *Read in nonconsecutive order.*
[d] *Write nonconsecutive keys.*
[e] *Read nonconsecutive keys, in different order than written.*
[f] *Read as above, change one of two keys, rewrite.*

A sequential I/O accesses records in order, while a direct I/O accesses records at random. An indexed I/O obtains a key from a key file and then it is able to access the desired record in the main file using that key.

of increasing complexity, each with full listings, explanations, and suggested exercises for the user to try. The programs are supplied on disk in both source and compiled form. This teaching method is most effective for the user who has experience with programming but not with this language.

Although the examples demonstrate many of the features of the language that are not explained elsewhere, they are not an adequate substitute for a complete language manual. To be beneficial, explanations and examples in program fragments should be incorporated in the reference manual.

HARDWARE PROTECTION SYSTEM

Perhaps the most controversial aspect of Clarion is its hardware-based protection scheme. The actuator as a means of protection certainly has some advantages. It is not a means of *copy* protection, as it allows unlimited copies to be made. Clarion can be installed on a hard disk and does not need to be deinstalled when restoring a disk from backup media. The possession of an actuator is the only authorization needed for making use of new distributions.

The scheme works best if Clarion and its actuator are installed on a system and never moved. In real life, however, actuators need to be inserted and removed. If only one actuator is available, the user will have to move it from system to system.

The actuator is required for running applications. In the end-user environment, where there is even less control over the system an application runs on, the developers using Clarion will be saddled with whatever problems their users eventually run into.

Clarion also checks for the actuator periodically throughout execution. If it fails to detect the actuator at the start-up of any Clarion utility, the system returns to DOS. However, if the actuator is removed during execution (presumably in an attempt to start a second copy on another machine) the computer hangs without any message or warning to the user whatsoever. Any unsaved work is sent into the bit bucket, or worse, unclosed data files can become corrupted. Unplugging printer cables, an all too common occurrence wherever the computers outnumber the printers, may remove or loosen the actuator sufficiently to cause problems.

Each potential user has to weigh the pros and cons of the protection system. No recommendation on that basis can be made here.

Despite a few flaws, notably in mapping modular programs, Clarion is a successful language. Its concept is boldly conceived, and its design has elegance and power, especially in programming console I/O.

The major fault of the development environment is the large size of all the components. Clarion is very robust; no bugs were found, and the system could not be made to crash. All of the criticisms leveled in this review address problems in the design, not errors in implementation. Overall, the design of the product ably addresses its stated aim, that of providing the application developer with the tools for rapid creation of salable products.

On its own merits—especially to those who judge its execution-protection scheme to be harmless, acceptable, or at least tolerable—Clarion can be highly recommended.

Clarion: $295
Barrington Systems, Inc.
P.O. Box 5580
Pompano Beach, FL 33074
800/354-5444
CIRCLE 339 ON READER SERVICE CARD

Ted Mirecki is a corporate planner who is responsible for developing decision support systems on a variety of hardware.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 66 BOULDER, CO.

POSTAGE WILL BE PAID BY ADDRESSEE

TECH PC JOURNAL
P.O. 2966
Boulder, Colorado 80321

LISTING 1: CLBENCH.CLA

```
BENCH        PROGRAM
             MAP
               PROC(SIEVE)
               PROC(MULDIV)
               PROC(SEQIO)
               PROC(DIRIO)
               FUNC(GETN),LONG
               MODULE('BENCHX')
                 PROC(KEYIO)
             . .

TEST_MENU    SCREEN       HUE(7,0,0)

               ROW(7,26)  PAINT(1,29),HUE(15,0)

               ROW(7,27)  STRING('CLARION EXECUTION BENCHMARKS')
               ROW(10,28) MENU,REQ
                 COL(30)    STRING('SIEVE OF ERATOSTHENES')
               ROW(12,30)   STRING('MULTIPLY & DIVIDE')
               ROW(14,30)   STRING('SEQUENTIAL FILE I/O')
               ROW(16,30)   STRING('DIRECT FILE I/O')
               ROW(18,30)   STRING('KEYED FILE I/O')
               ROW(20,30)   STRING('EXIT TO SYSTEM')
             .            .
TIME1        LONG
TIME2        LONG
ELAPSED      REAL
TIMESHOW     GROUP
               STRING('ELAPSED TIME:')
TIMEPIC        STRING(@N10.1)
               STRING(' SECONDS')
             .

  CODE
  LOOP
    OPEN(TEST_MENU)
    ACCEPT
    CLOSE(TEST_MENU)
    EXECUTE CHOICE()
      SIEVE
      MULDIV
      SEQIO
      DIRIO
      KEYIO
      RETURN
    .
    ELAPSED = (TIME2 - TIME1)/100.0
    TIMEPIC = ELAPSED
    SHOW(23,5,TIMESHOW)
    SHOW(24,5,'PRESS A KEY TO CONTINUE')
    ASK
SIEVE        PROCEDURE

LIM          SHORT(8191)
FLAGS        SHORT,DIM(8191)
K            SHORT
I            SHORT
PRIME        SHORT
COUNT        SHORT(0)
MSG          GROUP
COUNTPIC       STRING(@N10)
               STRING(' PRIMES')
             .

  CODE                                       !START EXECUTING SIEVE
  SHOW(2,5,'PRIME NUMBER SIEVE')
  TIME1 = CLOCK()                            !GET THE TIME IN 1/100 SECS
  LOOP I = 1 TO LIM
    FLAGS[I] = -1
  .
  LOOP I = 1 TO LIM
    IF FLAGS[I] <> 0
      PRIME = I+I+3
      K = I+PRIME
      LOOP WHILE K <= LIM
        FLAGS[K] = 0
        K = K + PRIME
      .                                      !END OF WHILE LOOP
      COUNT = COUNT + 1
  . .                                        !END OF IF & LOOP TO LIM
  TIME2 = CLOCK()
  COUNTPIC = COUNT
  SHOW(4,5,MSG)

MULDIV       PROCEDURE

N            SHORT(5000)                     !LOOP COUNT
A            REAL(2.71828)
B            REAL(3.14159)
C            REAL(1.0)
MSG          GROUP
               STRING(' ERROR ')
ERRPIC         STRING(@E11.1)
             .

  CODE
  SHOW(2,5,'MULTIPLICATION - DIVISION BENCHMARK')
  TIME1 = CLOCK()
  LOOP N TIMES
    C = C * A
    C = C * B
    C = C / A
    C = C / B
  .
  TIME2 = CLOCK()
  ERRPIC  = 1.0 - C
  SHOW(4,5,MSG)

GETN         FUNCTION
NRECS        LONG

  CODE
  SHOW(2,5,'NUMBER OF RECORDS?')
  ASK(2,24,NRECS,@N4)
  BLANK
  RETURN(NRECS)

SEQIO        PROCEDURE

NRECS        SHORT
I            SHORT
MSG          GROUP
               STRING('SEQUENTIAL I/O - ')
NRECPIC        STRING(@N4)
               STRING(' RECORDS')
             .
SEQFIL1      FILE,NAME('A:\SEQFIL1.DAT')
SEQREC1        RECORD,PRE(IN)
LABEL1           STRING(7)
RECNUM           STRING(@N4)
TAIL             STRING(89)
             . .                             !END OF RECORD & FILE DEF

SEQFIL2      FILE,NAME('A:\SEQFIL2.DAT')
SEQREC2        RECORD,PRE(OUT)
LABEL1           STRING(7)
RECNUM           STRING(@N4)
TAIL             STRING(89)
             . .                             !END OF RECORD & FILE DEF

  CODE
  NRECS = GETN()
  NRECPIC = NRECS
  SHOW(2,1,MSG)
  SHOW(4,5,'WRITING...')
  TIME1 = CLOCK()
  STREAM(SEQFIL1)
  BUFFER(SEQFIL1,0.5)
  LOOP I = 1 TO NRECS
    IN:LABEL1 = 'RECORD'
    IN:RECNUM = I
    ADD(SEQFIL1)
    CLOSE(SEQFIL1)
    TIME2 = CLOCK()
```

```
    ELAPSED = (TIME1 - TIME2)/100.0
    TIMEPIC = ELAPSED
    SHOW(4,17,TIMESHOW)

    SHOW(6,5,'READING...')
    TIME2 = CLOCK()
    BUFFER(SEQFIL1, 0.5)
    SET(SEQFIL1)
    LOOP NRECS TIMES
      NEXT(SEQFIL1)
    .
    CLOSE(SEQFIL1)
    ELAPSED = (CLOCK() - TIME2)/100.0
    TIMEPIC = ELAPSED
    SHOW(6,17,TIMESHOW)

    SHOW(8,5,'COPYING...')
    TIME2 = CLOCK()
    BUFFER(SEQFIL1,0.5)
    SET(SEQFIL1)
    STREAM(SEQFIL2)
    BUFFER(SEQFIL2,0.5)
    LOOP UNTIL EOF(SEQFIL1)
      NEXT(SEQFIL1)
      SEQREC2 = SEQREC1
      ADD(SEQFIL2)
    .

    CLOSE(SEQFIL1)
    CLOSE(SEQFIL2)
    ELAPSED = (CLOCK() - TIME2)/100.0
    TIMEPIC = ELAPSED
    SHOW(8,17,TIMESHOW)
    TIME2 = CLOCK()


DIRIO       PROCEDURE

NRECS       SHORT
I           SHORT
HALFWAY     SHORT
RECPTR      SHORT
MSG         GROUP
              STRING('DIRECT I/O - ')
NRECPIC       STRING(@N4)
              STRING(' RECORDS')
            .
DIRFIL1     FILE,NAME('A:\SEQFIL1.DAT')
DIRREC1       RECORD
LABEL1          STRING(7)
RECNUM          STRING(@N4)
TAIL            STRING(89)
            . .                                       !END OF RECORD & FILE DEF

    CODE
    NRECS = GETN()
    NRECPIC = NRECS
    SHOW(2,1,MSG)
    SHOW(4,5,'READING...')
    TIME1 = CLOCK()
    BUFFER(DIRFIL1,0.5)
    HALFWAY = NRECS/2 + 1
    LOOP I = HALFWAY TO NRECS
      RECPTR = I
      GET(DIRFIL1, RECPTR)
      RECPTR = I - HALFWAY + 1
      GET(DIRFIL1, RECPTR)
    .
    CLOSE(DIRFIL1)
    TIME2 = CLOCK()
    ELAPSED = (TIME1 - TIME2)/100.0
    TIMEPIC = ELAPSED
    SHOW(4,17,TIMESHOW)

    SHOW(6,5,'UPDATING...')
    TIME2 = CLOCK()
    STREAM(DIRFIL1)
    BUFFER(DIRFIL1, 0.5)
    HALFWAY = NRECS/2 + 1
    LOOP I = HALFWAY TO NRECS
```

```
  RECPTR = I
  GET(DIRFIL1, RECPTR)
  PUT(DIRFIL1)
  RECPTR = I - HALFWAY + 1
  GET(DIRFIL1, RECPTR)
  PUT(DIRFIL1)
.
CLOSE(DIRFIL1)
ELAPSED = (CLOCK() - TIME2)/100.0
TIMEPIC = ELAPSED
SHOW(6,17,TIMESHOW)
TIME2 = CLOCK()
TIME2 = CLOCK()
```

LISTING 2: CLINDEX.CLA

```
              MEMBER('BENCH')

KEYIO         PROCEDURE

NRECS         SHORT
I             SHORT
J             SHORT
HALFWAY       SHORT
MSG           GROUP
                STRING('KEYED I/O - ')
NRECPIC         STRING(@N4)
                STRING(' RECORDS')
              .
KEYFIL1       FILE,NAME('A:\KEYFIL1.DAT')
MAINKEY         KEY(RECNUM)
ALTKEY          KEY(ALTNUM), DUP
KEYREC1         RECORD
LABEL1            STRING(7)
RECNUM            STRING(@N4)
LABEL2            STRING(5)
ALTNUM            STRING(@N4)
TAIL              STRING(80)
              . .                                 !END OF RECORD & FILE DEF

  CODE
  NRECS = GETN()
  NRECPIC = NRECS
  SHOW(2,1,MSG)
  SHOW(4,5,'WRITING...')
  TIME1 = CLOCK()
  STREAM(KEYFIL1)
  BUFFER(KEYFIL1,0.5)
  HALFWAY = NRECS/2 + 1
  J = 0
  LOOP I = HALFWAY TO NRECS
    LABEL1 = 'RECORD'
    RECNUM = I
    LABEL2 = ' ALT '
    J += 1
    ALTNUM = J
    ADD(KEYFIL1)
    LABEL1 = 'RECORD'
    RECNUM = I - HALFWAY + 1
    LABEL2 = ' ALT '
    J += 1
    ALTNUM = J
    ADD(KEYFIL1)
  .
  CLOSE(KEYFIL1)
  TIME2 = CLOCK()
  ELAPSED = (TIME1 - TIME2)/100.0
  TIMEPIC = ELAPSED
  SHOW(4,17,TIMESHOW)
  SHOW(6,5,'READING...')
  TIME2 = CLOCK()
  BUFFER(KEYFIL1, 0.5)
  CACHE(MAINKEY, 0.5)
  J = 0
  LOOP I = NRECS TO HALFWAY BY -1
    RECNUM = I
    GET(KEYFIL1, MAINKEY)
    J += 1
    RECNUM = J
    GET(KEYFIL1, MAINKEY)
  .
CLOSE(KEYFIL1)
ELAPSED = (CLOCK() - TIME2)/100.0
TIMEPIC = ELAPSED
SHOW(6,17,TIMESHOW)

SHOW(8,5,'UPDATING...')
TIME2 = CLOCK()
BUFFER(KEYFIL1, 0.5)
CACHE(MAINKEY, 0.5)
J = 0
LOOP I = NRECS TO HALFWAY BY -1
  RECNUM = I
  GET(KEYFIL1, MAINKEY)
  ALTNUM = ALTNUM + HALFWAY
  PUT(KEYFIL1)
  J += 1
  RECNUM = J
  GET(KEYFIL1, MAINKEY)
  ALTNUM = ALTNUM + HALFWAY
  PUT(KEYFIL1)
.
CLOSE(KEYFIL1)
ELAPSED = (CLOCK() - TIME2)/100.0
TIMEPIC = ELAPSED
SHOW(8,17,TIMESHOW)

TIME2 = CLOCK()
```

Before you consider the new Hercules Graphics Card Plus, consider the technology behind it.

On July 1, Hercules™ introduced a product that will forever change the way information is displayed on a PC.

The product is called the Hercules Graphics Card Plus.

We gave it that name because it gives you the same hi-resolution text and 720x348 graphics that made the original Hercules Graphics Card famous.

Plus it gives you RamFont.™

RamFont is a radical new hardware mode that combines the speed of text mode with the flexibility of graphics mode.

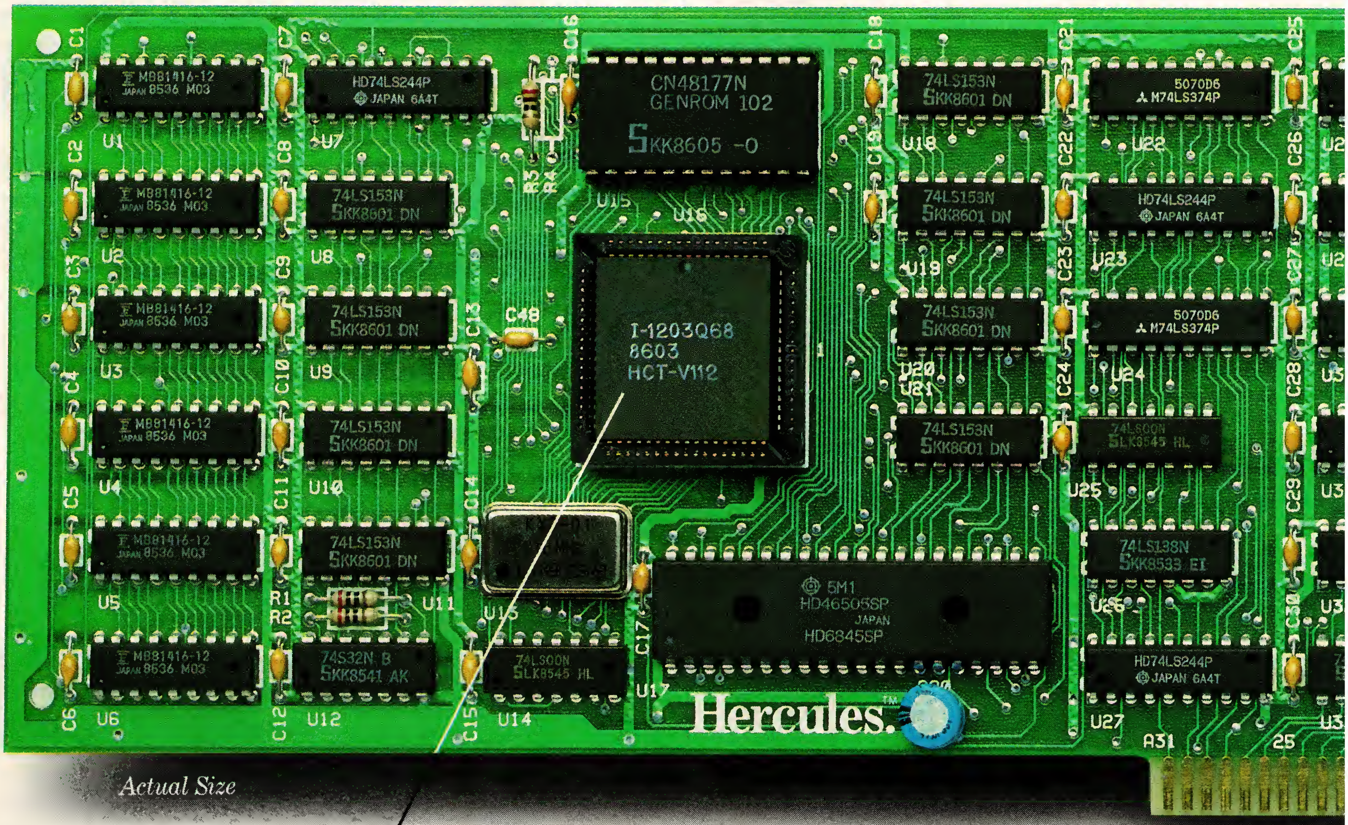
And opens up a whole new world for software.

The world according to RamFont.

In the old days (before July 1), programs like Lotus® 1-2-3®, Symphony,™ Framework™ and Microsoft® Word had to use graphics mode to display multiple fonts and variable text sizes, or to mix text with graphics.

But graphics mode is a whole lot slower than text mode. Up to eight times slower.

Enter RamFont.



Actual Size

The heart of the Graphics Card Plus: the V112 microchip, Hercules' next generation video processor that makes the RamFont mode possible.

It lets all the programs we just mentioned (and plenty more in the future) do all the things we just described, all on one screen.

At precisely the same speed as text mode.

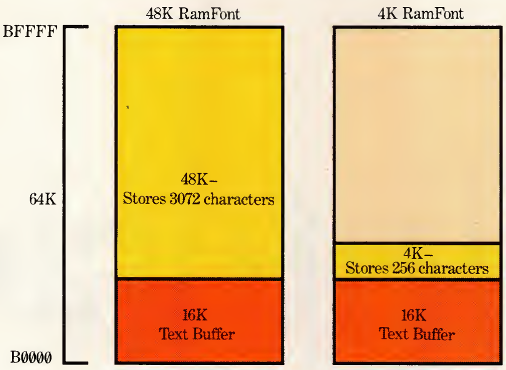
How RamFont works.

Like text mode, RamFont uses a 16-bit word to represent a character on the display.

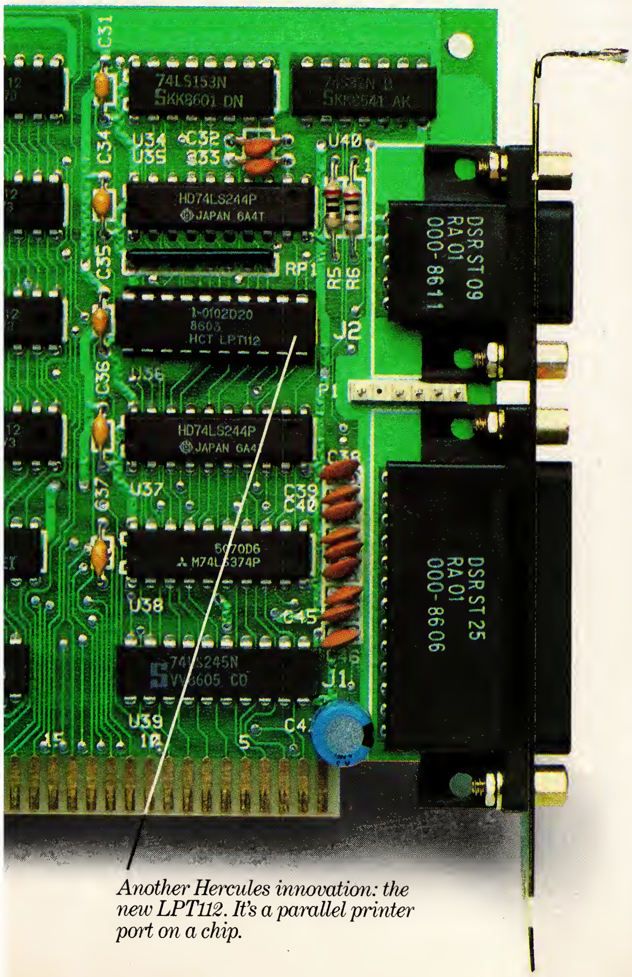
Unlike text mode, however, the 48K RamFont mode uses a 12-bit character code instead of an 8-bit code.

Which allows you to choose from an astonishing 3072 different characters.

While setting the size of your screen cells from eight to nine pixels wide and from four to 16 scan lines tall.



The new RamFont mode displays 3072 programmable characters at the speed of text mode, or replaces the standard character set with one of your choice in 4K RamFont mode.



Another Hercules innovation: the new LPT112. It's a parallel printer port on a chip.

To help you design your own RamFont characters and symbols, we've included a font editor called FontMan.™

Along with a set of 25 sample fonts to start your library.

And since the 4K RamFont mode can accept 8-bit character codes, running your text mode software with your favorite font is as simple as loading it into RAM.

Just for the record, RamFont supports the standard character attributes of reverse, high-intensity, blink and underline.

Plus two new RamFont attributes: boldface and strike-through.

What price success?

While we're on the subject of technological breakthroughs, let's not forget the suggested retail price.

Think of it. You get everything the original card gave you.

Plus RamFont.

Plus FontMan.

Plus a free printer cable.

Plus a parallel printer port that you can now disable if there's a conflict with another port in your system.

All for just $299—about half the price of the original Graphics Card.

If you haven't gone into shock, call 1-800-532-0600 ext. 211 for the name of an Authorized Hercules Dealer near you and we'll rush you our free info kit.

$299.
Suggested retail price.

Hercules.™

We're strong on graphics.

Address: 2550 Ninth Street, Berkeley, California 94710 Ph: 415 540-6000
Telex: 754063

CIRCLE NO. 242 ON READER SERVICE CARD

Interactive CAD

The VersaCAD system adds an on-line database management facility to a micro-based CAD package.

VICTOR E. WRIGHT

Whenever a group of microcomputer CAD users assembles, a discussion is certain to ensue about which CAD program is the best. One product likely to have a group of loyal supporters is VersaCAD, from T & W Systems, Inc. This company pioneered microcomputer CAD with the release of its first system in early 1981. VersaCAD was introduced in January 1983, only weeks after AutoDesk introduced AutoCAD (for a review of AutoCAD, see "Drafting by Design," Victor E. Wright, January 1986, p. 50). Although VersaCAD has not matched AutoCAD's sales volume, the program has a substantial following.

Part of its continued success may be attributed to VersaDATA, an optional module that adds interactive database management capabilities to the basic CAD package. VersaDATA goes one step farther than most microcomputer CAD packages by providing an on-line database management facility with an input and report form capability for the attachment of nongraphic attributes to drawing objects. Input forms and report forms are displayed on text screens developed from within VersaDATA during the drawing process. Changing the graphic properties of an object on the screen changes them immediately on the form. Changing the same properties on the form produces an immediate effect on the graphic display.

The full VersaCAD line, which runs on the IBM PC, PC/XT, PC/AT, and compatibles as well as the 32-bit Hewlett-Packard Series 200 minicomputer, includes VersaCAD Entry Level, a low-cost program aimed at the educational market; VersaCAD Professional, a mid-range drafting-only program; VersaCAD Advanced (version 4.0), a production drafting system with nongraphic database management and three-dimensional modeling capabilities; and VersaCAD Advanced (version 5.0), an open architecture two-dimensional drafting program that can be customized. VersaCAD 5.0 is currently a companion rather than a replacement for version 4.0.

Both versions of the VersaCAD Advanced system are reviewed here, along with VersaDATA and VersaLIST, a module that generates bills of material automatically. The full system also includes VersaLINK, a module that converts a VersaCAD drawing to ASCII form, and a 3-D wire frame modeling program. The basic VersaCAD package is furnished on five 5¼-inch floppy disks. The modules add seven more disks to the package, and the documentation fills an 8½-by-11-by-2-inch thick loose-leaf binder.

VersaCAD 4.0. requires 384KB of memory. The newer version 5.0 requires 640KB, leaving insufficient memory to run the interactive VersaDATA module. T & W Systems anticipates that version 5.0 will support VersaDATA by late September, using some form of expanded memory.

The basic VersaCAD drawing world is a two-dimensional plane. Although it

COMPUTER GRAPHIC • NATALE/EAST

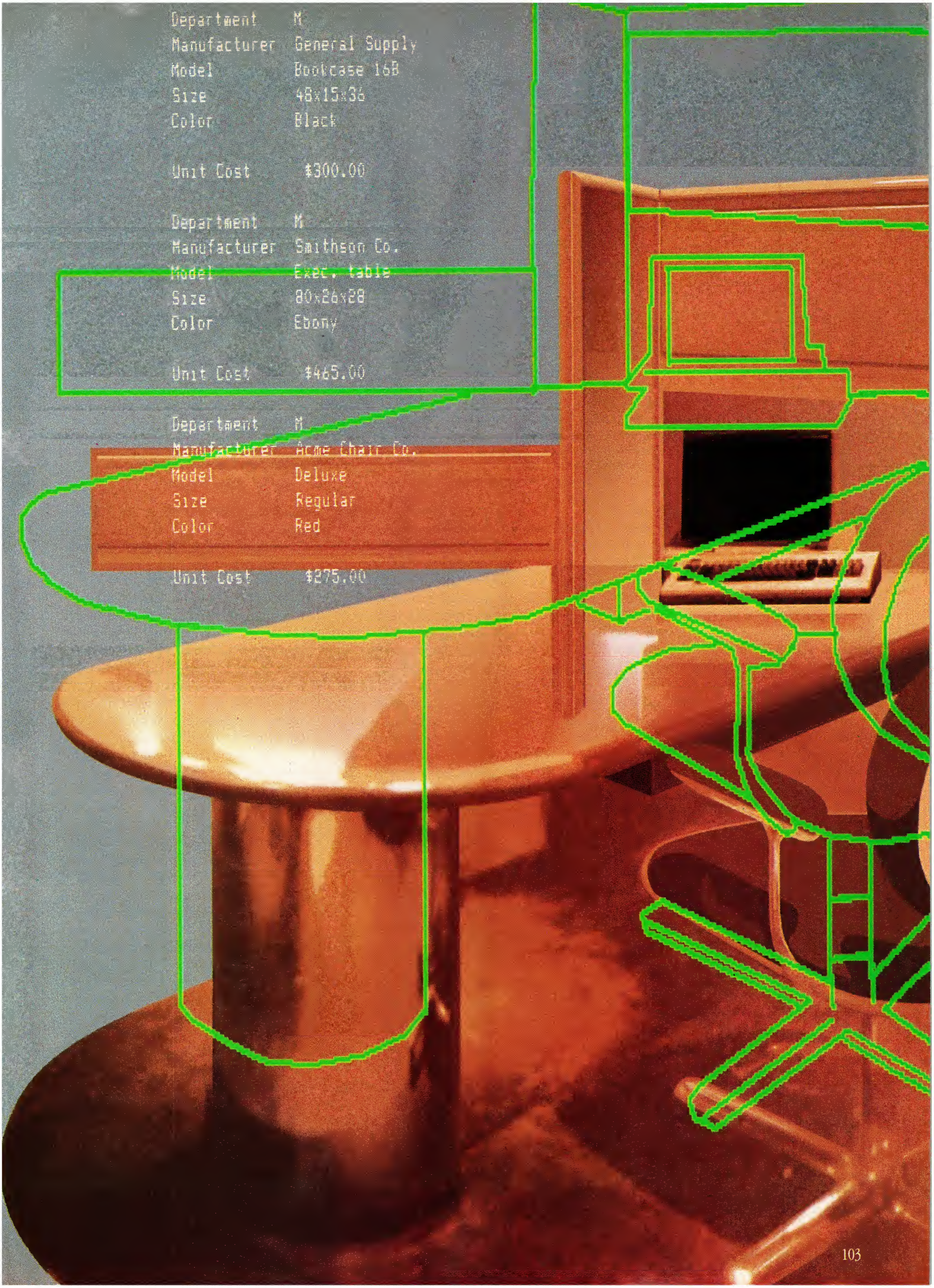
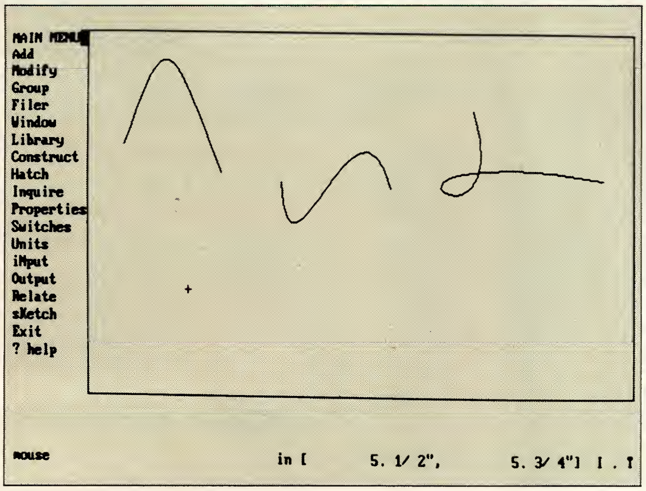
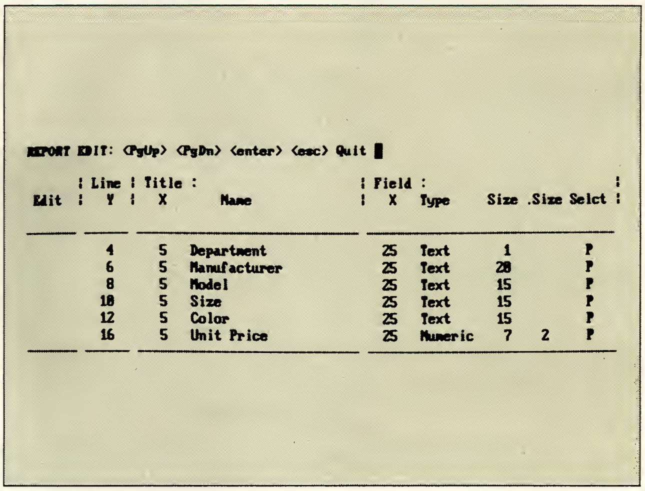

Department M
Manufacturer General Supply
Model Bookcase 16B
Size 48x15x36
Color Black
Unit Cost $300.00
Department M
Manufacturer Smithson Co.
Model Exec. table
Size 80x26x28
Color Ebony
Unit Cost $465.00
Department M
Manufacturer Acme Chair Co.
Model Deluxe
Size Regular
Color Red
Unit Cost $275.00

PHOTO 1: *Menu and Bezier Curves*



The screen is split into the menu area, drawing display window, and a status line. The default mode of the Bezier option creates a curve with no inflection point. These curves also can be S-shaped or have a loop between endpoints.

PHOTO 2: *Example of Report Editing*



The required output form of a report in VersaDATA can be customized according to the user's requirements. The *X* and *Y* coordinates for any of the selected fields or titles can be modified, or the titles themselves can be changed.

is not unlimited, or even as large as those of some of its competitors, it is certainly adequate at 1.96^{107} units in the *X* direction and 7.2^{106} units in the *Y* direction. Units can be interpreted and displayed as inches, feet, meters, kilometers, miles, or userdefined measurements. Drawings are limited to approximately 15,000 objects. The exact drawing file size in bytes is determined by the types of objects used.

The program accepts input with seven digits to the right of the decimal point, although it displays only four places, or fractions of inches to one-sixty-fourth. This should provide adequate precision for most calculations. Internally, VersaCAD performs calculations with eleven decimal digits of precision.

VersaCAD was tested on three hardware configurations: a PC with 640KB RAM, 8087 math coprocessor, 15MB hard disk, Hercules Graphics Card, and Hitachi Tiger digitizer; an AT with 640KB RAM, 80287, 30MB hard disk, IBM Enhanced Graphics Adapter with 256KB video RAM, and Kurta Series 2 12-by-17-inch digitizer; and a Heathkit H-200 AT compatible with 640KB RAM, 80287, 20MB hard disk, ConoGraphic ConoColor 40 graphics card, Houston Instrument True Grid 24-by-18-inch digitizer. All three systems were connected to a PRIAM SharedSpace disk drive. A Houston Instrument DMP-52 MP plotter was used for plotting tests. A Princeton Graphic Systems HX-12E was used as the display in each of the configurations. VersaCAD supports dual screen configurations.

The program is not copy protected, so installation is a simple matter of copying the files from the distribution disks to the working disk. Plenty of hard-disk space is required; at the conclusion of the installation process during testing, a DIR command revealed that the system had taken up more than 3MB, even though only the required device drivers were transferred.

Because the program makes effective use of path names to store its various data files, it is advisable to create a directory for the main program files and subdirectories for storage of the data files. To facilitate this process, the distribution disks include batch program files that automate the installation process to the point of creating the directories and subdirectories.

The INSTALL batch program calls the configuration program (called ENVIRO40.EXE) if it determines the program has not been configured previously. The configuration file created by ENVIRO40 contains information on both the hardware and the subdirectory structure. Although the INSTALL program creates a default subdirectory structure, the ENVIRO40 program can modify that structure.

The basic VersaCAD Advanced program, VersaDATA, and the 3-D modeling program are each furnished with a complete set of device drivers from which ENVIRO40 constructs the configuration file. The manual supplies clear information for customizing the batch file used to start the program to ensure that the necessary drivers are supplied. For example, to use the Mouse Systems mouse on COM2, the line MSMOUSE/2 must be added to the batch file.

During the configuration process for this review, the Kurta digitizer was connected and configured as outlined in the manual, but did not work. The documented configuration used only a 3-inch wide by 2-inch high area of the digitizer as an active area for screen pointing and assumed low resolution; the tablet on hand was high resolution. The solution was to enter the digitizing surface dimensions as 85-by-60 inches during the configuration process.

The user interface for VersaCAD comprises windows on the display screen. The graphics window, where the drawing is displayed, is the largest. On the Princeton HX-12E, this area measures 7½ inches wide by 5 inches high; on an IBM monitor it would be slightly larger. This window occupies the upper right corner of the screen.

To the left of the graphics window is the menu area, which is 10 characters wide and 21 lines high. Immediately below the menu and graphics windows is a three-line area where messages and prompts are displayed. At the bottom of the screen is a status line that displays the input mode (input device and type of coordinate display), the current cursor coordinates, the snap mode, and the status of object tracking (see photo 1).

VersaCAD is organized around the main menu and submenus to various levels, depending upon the command. At any given time, the main menu or a submenu is displayed in the menu window, and commands on the current displayed menu can be entered by

LOOKING FOR THE BEST VALUE IN PC/AT COMPATIBLES?

INTRODUCING THE $1495 A★STAR™

Looking for a lower priced, higher quality PC/AT compatible than our new A★Star™ is like looking for a needle in a haystack. You're more apt to come up with a case of hayfever.

A★Star gives you features you won't get from *any* other vendor. Not IBM. Not Compaq. Not even those foreign manufacturers. Features like a 220 watt power supply, eight *available* expansion slots and "network ready" multi-user operation. All for only $1495. And that price includes a 1.2MB diskette drive, 512KB memory, a fixed disk/diskette drive controller, a tactile feedback keyboard (you'll love it!) and a 6/8MHz switch/software selectable 80286 CPU.

Best of all, A★Star is quality built right here in America by Wells American Corporation - an American Stock Exchange manufacturer. So now you can buy with confidence from a vendor you can trust, just like you've trusted IBM. And you'll get a genuine, top quality product (not a cheap imitation) for a price even less than those "questionable" mail order clones. And if that's not enough, it's all money-back guaranteed!

The $1495 Wells American A★Star.™

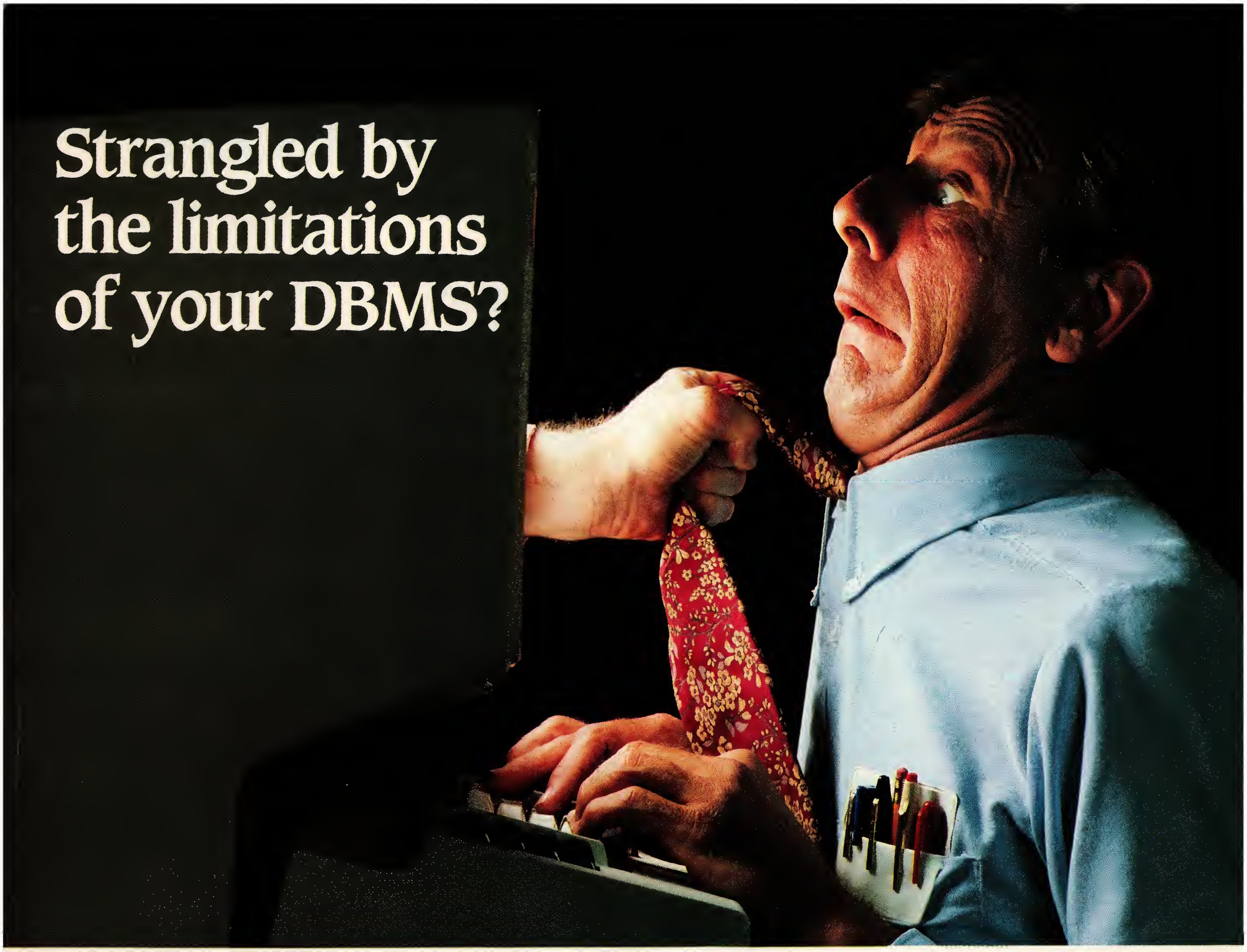
A PC/AT FOR ONLY $1495? THAT'S NOTHING TO SNEEZE AT!

(Hurry! This offer *is* limited.)

- ☐ I'm tired of sneezing! Have someone call me immediately to take my order.
- ☐ A★Star sounds terrific. Tell me more.

Name: ____________________
Company: ____________________
Phone: ____________________
Address: ____________________
City: ____________________
State: ____________ Zip: ________

Wells American
803/796-7800•TWX 510-601-2645
Sunset Boulevard • West Columbia, SC 29169



ZIM. The system for serious developers

Your data base system was easy to learn. You were building your first simple applications within hours. You have a menu driven interface and pop up menus with online context sensitive help. Now you are ready to build your first real application. Instant Strangulation!

The Symptoms

- There is no real 4th Generation Language — what you see at the menu driven level is what you get.
- There are all sorts of built-in limitations which no real problem ever obeyed
- Your database which runs like a superstar on a few files and 500 records, chokes when faced with several files of 50,000 records
- You cannot efficiently run your applications in UNIX, VAX VMS and MS-DOS without major surgery, or finding a new DBMS
- Your DBMS supplier asks, "what is networked multiuser anyway?"
- Your database is too big to live on one disk but your DBMS doesn't support multiple disks
- Your database can't answer simple questions like "tell me which inventory items have **NOT** sold in the past 3 months?"
- You built your application using the nice easy interface, but now it won't get out of the way to allow you to modify the application
- You can't compile your applications and there is no runtime system

Diagnosis

Extreme stress and frustration in your development staff and applications crippled at birth.

THE CURE

ZIM is the 4th Generation Language/DBMS for serious application developers. ZIM is a top performer in all of its environments (UNIX, XENIX, MS-DOS, Novell Networks, VAX VMS, QNX), for all types of information application. ZIM's easy menu driven interface with context sensitive help is not meant just for novice users, and it stays discreetly out of the way when it isn't needed.

You need never suffer the symptoms of Database strangulation again.

ZANTHE INFORMATION INC.
1200-38 Antares Drive
Nepean, Ontario, Canada K2E 7V2
Sales (US) 800-267-9972 (CDN) 613-727-1397

ZANTHE INFORMATION INC

ZIM™

CIRCLE 219 ON READER SERVICE CARD

pressing a single capital character or by using a pointing device to place the menu cursor on the desired command and pressing the pick button.

The pointing device controls the cursor's position, which is displayed as a cross when positioned in the graphics window and as a reverse video bar in the menu area. VersaCAD allows the user to position the cursor off-screen with the pointing device, an irritating quirk. This means more accuracy than necessary is required to position the cursor on a menu item.

Most commands must be entered from the correct menu or submenu, requiring that the somewhat rigid menu structure be navigated. Certain commands can be entered with function keys, however. Eight of the ten function keys are operational: F1 (input mode) selects the type of data input; F2 (snap mode) cycles the snap modes among the four options—increment snap, grid snap, object snap, and snap off; F3 (reference/origin) allows the origin to be repositioned or current cursor location to be selected for relative or polar data input; F4 (object tracking) toggles the object-tracking mode, which continuously redraws objects when they are being positioned; F5 (cursor size) toggles the cursor between a small cross and a cross that extends across the graphics window; F6 (update) allows specification of the current value of various global properties; F7 (device reset) sets pointing device movement to the default scale; and F8 (device scale) scales the movement of the pointing device to provide for more precise control of the screen cursor.

FILE MANAGEMENT

VersaCAD creates a drawing in a temporary file called a work file, found between drawing sessions in the VCAD directory under the name VCAD40.WRK. Permanent drawings are placed in a subdirectory; the default subdirectory is VCAD\DRAW. The work file can be retained from one drawing session to the next, but only one copy exists at any given time. Permanent drawing files must be explicitly saved with the Save option under the FILER menu.

In addition to the work file, VersaCAD maintains a temporary library file called VCAD40.LIB that contains the symbol definitions used in the drawing. The contents of VCAD40.WRK and VCAD40.LIB are merged into a single, permanent drawing file using the FILER Save command.

The FILER command provides two options to retrieve drawing files. The Get option clears the current work file and retrieves a copy of the specified drawing file; the original file remains intact. The Merge option retrieves a copy of the specified drawing file, but inserts it into the current work file without deleting the information already stored there. Merge provides a type of symbol insertion capability in that a composite drawing can be created from several other drawing files; the merged drawing does not remain a single identity, however.

Other FILER options clear the work file, delete drawing files from disk, crunch (or compress) the work file, list currently saved drawings, display a summary of the usage of the current work file, back up the work file, and change the default drawing storage file.

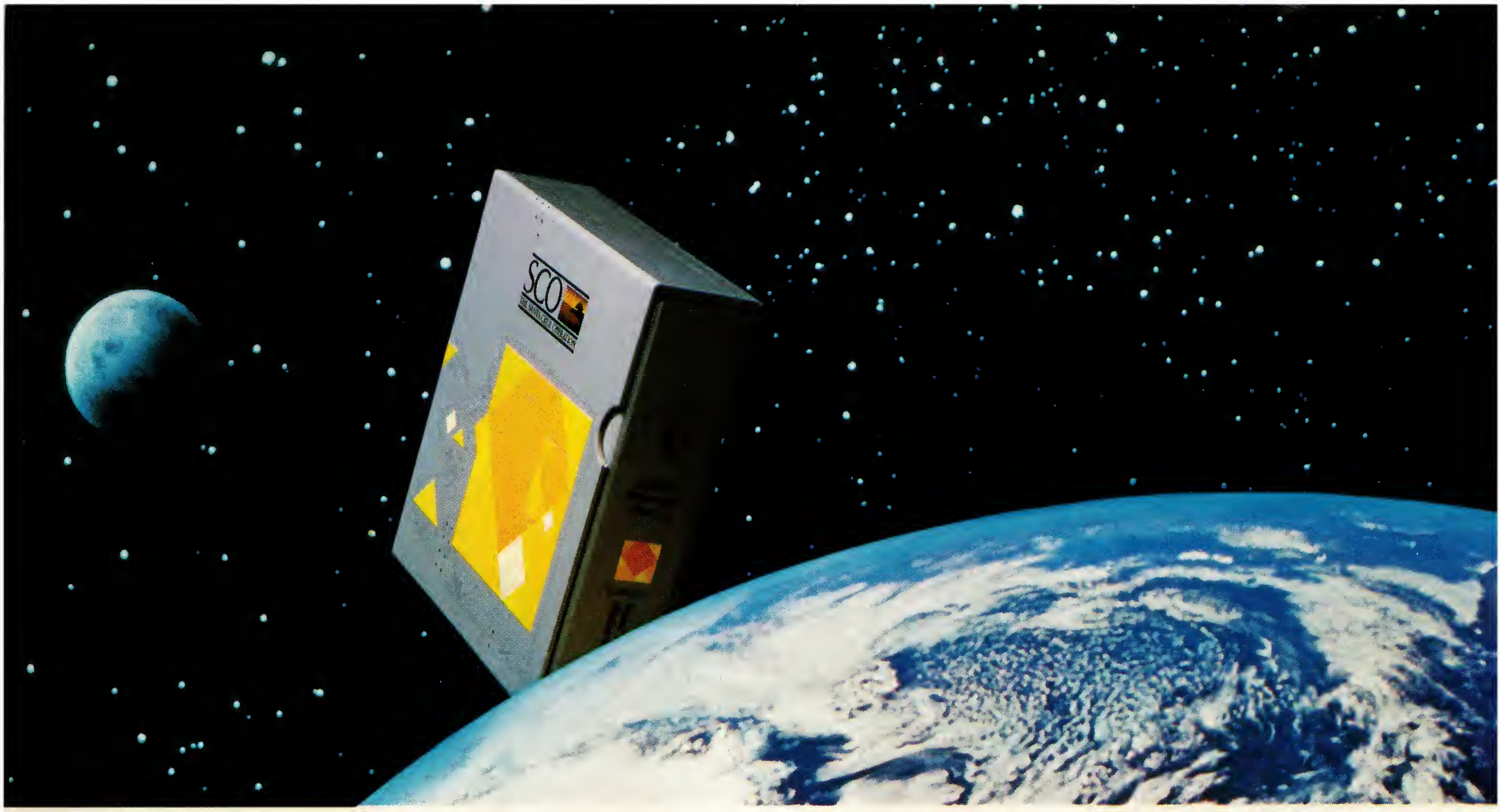
The FILER Ext list option lists drawings by name, but also includes several useful statistics for each drawing: date and time last modified, cumulative drawing time, object total, and symbol total. Used periodically and printed with the Shift-PrtSc keystroke, this command provides the information needed for client billing or internal cost accounting. The What option provides similar information about the work file.

ZOOM IN, ZOOM OUT

VersaCAD includes a complete assortment of display controls, most of which are provided in the WINDOW menu. Although the entire drawing universe cannot be displayed on the screen at once, a zoom capability allows shrinking or enlarging the display sufficiently for any reasonable application. VersaCAD uses the terms Window In and Window Out instead of *zoom* in the menu. Both commands allow the new display to be specified by pointing to opposite corners of the desired area.

Both Window In and Window Out allow the zoom window to be panned before the display is redrawn at the new magnification ratio. The Pan command moves the graphics window without changing the scale. VersaCAD works with the paradigm of moving the window across a stationary drawing, rather than that of moving the drawing under a stationary window.

Another zoom command, Base, returns to a preset screen that is defined via the Base option of the UNITS command. This helps users regain their bearings after windowing in and out several times in succession. The option WINDOW Full performs a similar function, displaying the smallest window that can hold all the objects currently contained in the drawing.



XENIX RISING

"It was back in 1986 that we first grasped the full significance of **SCO XENIX**®.

"We had been thinking of it as just an 'operating system'—as we used to call them—when it was really much more. It was the foundation of a whole new approach to shared information and resource computing for PCs: networked DOS *and* XENIX workstations.

"The '**SCO XENIX** solution' integrated XENIX and DOS, multiuser and LAN, and PCs and mainframes, into a unified environment unprecedented in its power, productivity, and price performance per user for personal computers.

"SCO's solution included not only 16-user licensed **SCO XENIX** itself, but also **SCO XENIX-NET** XENIX/DOS networking; **SCO Professional**™ and **SCO FoxBASE**™, SCO's XENIX-based workalikes of 1-2-3® and dBASE II™; **SCO Lyrix**™ Word Processing System, and other productivity tools; and **SCO uniPATH**™ SNA-3270 Mainframe Communications, plus languages and graphics packages.

"Together with unparalleled SCO hot-line support, documentation, and training, plus the explosive growth of the XENIX applications base, it changed the way we would look at personal computers forever.

"In short, it was the '**SCO XENIX** solution' that turned personal computers into *real* computers."

from *"XENIX RISING:*
The Ascent of Shared Information
Computing from 1979 to 1999"

SCO SALES AND INFORMATION
(800) 626-UNIX
(800) 626-8649
(408) 425-7222
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info

SCO
THE SANTA CRUZ OPERATION

See us at Booth 121 at FCC, Washington D.C., Sept. 3–5

SCO XENIX System V is available for IBM® PC AT®, PC XT®, ITT XTRA® XL and other compatibles, and AT&T PC 6300 Plus–now!

CIRCLE NO. 121 ON READER SERVICE CARD

The WINDOW command provides options to Save, Get (retrieve), Delete, and List windows. The program does not actually save the screen image of a window, but a view definition, consisting of the coordinates and magnification of the view. Saved windows are referenced by names up to seven characters in length, including numbers.

Named windows are useful for moving around a drawing efficiently. For example, an electronics designer working on a large schematic might wish to name views of various areas—such as power supply, input/output, CPU—so that they can be recalled without windowing in or out.

TEN PRIMITIVES

VersaCAD provides ten drawing primitives: Line, Rectangle, Polygon, Ellipse, Circle, Arc, Bezier, Point, Dimension, and Text. These primitives can be entered from the ADD submenu, which is displayed with the ADD selection on the main menu.

Each primitive drawing command in the ADD menu displays a submenu of options appropriate to the particular command. The Line command string displays a menu with the choices Single, Erase, Quit, and Esc displayed below the header, LINE. Selecting Single means that single, isolated line segments can be entered; both the start and endpoints of each segment must be provided. If Single is not selected, line strings are entered—the endpoint of one segment is the start of the next.

In either case, when the start point is provided, by positioning the cursor and pressing the appropriate button on the pointing device, a second Line menu appears, and the user can specify the type of line to be entered. The options are Arrow, which places an arrowhead at the endpoint; Template, which creates a dotted line; Marker, which places an *X* at the midpoint of the current line segment; *X*-axis, which forces the current line segment into alignment with the *X*-axis; *Y*-axis, which does the same with the *Y*-axis; Free, which releases the *X*- or *Y*-axis lock; Rotate, which rotates the current line segment relative to the previous segment by a preset angle; Detach, which allows the entry of a new start point, breaking a line string; Single, which enters the Single line mode; Erase, which erases the last endpoint entered so the previous line segment can be repositioned; and Quit, which ends the Line command mode and returns to the ADD menu.

Effective use of this menu requires both the pointing device and the keyboard, because the screen cursor cannot be moved to the menu area during the entry of a line segment or string. A string of lines of various types—arrows, template lines, orthogonal lines—can be drawn quickly by entering points with the pointing device and selecting line types with the keyboard.

In addition, the function keys can be used within the Line menu for control of other line properties. F6 (update) displays a menu of properties: Line Style, Group Name, Density, Rotation, Line Width, Increment, Level, Text, Pen, Color, and Quit. Use of F6 requires several keystrokes in the midst of the Line command, but minimizes the need to edit properties at a later time.

VersaCAD includes some drawing primitives that are usually omitted from most other microcomputer-based CAD programs: rectangles, polygons, and ellipses. These are truly primitive drawing objects, not symbols that are constructed from line segments.

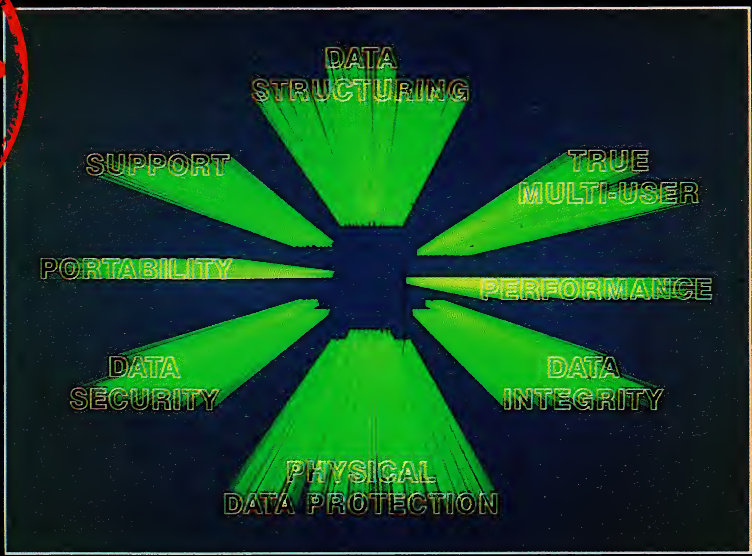
Circles are entered as expected, with the options of specifying the center and radius or the diameter. Arcs can be created in three ways: by specifying three points on the arc; two points on the arc and one defining its radius; or the center of the arc and two endpoints. An Arrow option places an arrowhead at the second endpoint of the arc; Free allows the user to specify the radius of the arc interactively; Radius allows the entry of the radius from the keyboard; and Direction controls the direction in which the arc is drawn when the center and two endpoints are specified. Template and Marker options also are available for use with arcs.

VersaCAD offers a "French curve" primitive with its Bezier option. Bezier curves provide a freehand sketching facility in an economical manner. Freehand sketching as implemented in some CAD programs requires large amounts of memory, because the freehand curves are composed of hundreds of short line segments. A Bezier curve consists of four data points—two endpoints and two control points—that the program uses to construct a smooth curve. The curve is tangent to a line between the first endpoint and the first control point, at the first endpoint, and to a line between the second control point and the second endpoint, at the second endpoint.

The default mode of the Bezier option is for both control points to follow the cursor and be positioned at the same location when the pick button is pressed on the pointing device, creating a curve with no inflection point. How-

MDBS III®

ATTN: APPLICATION DEVELOPERS



. . . DELIVERS THESE ESSENTIAL FEATURES. DOES YOUR DBMS?

MDBS III is more powerful than most mainframe data base management systems . . . and less expensive. MDBS III was designed for serious application developers like you. Like the developers of Solomon III, the "Number One" accounting system. And all the others who demand these essential features MDBS III provides:

DATA STRUCTURING—So flexible it captures any data relationship you can imagine. So comprehensive you'll design complex data bases faster than ever.

TRUE MULTI-USER—Few DBMSs give you as many facilities to guard against haphazard concurrent data modification as MDBS III does, down to the locking of individual data records.

PERFORMANCE—MDBS III gives you fast data modification and retrieval plus extensive performance tuning facilities.

DATA INTEGRITY—MDBS III provides airtight integrity assurances . . . from range checking to transaction-logging to enforcement of data relationships . . . *all automatically.*

PHYSICAL DATA PROTECTION—You get automatic recovery from media as well as from physical data destruction.

DATA SECURITY—Protect your data using passwords, encryption, and read/write access down to the field level.

PORTABILITY—MDBS III runs on a range of mini and micro computers, including LANs, and supports a variety of host language interfaces.

SUPPORT—**mdbs** is there when you need us, with in-depth seminars, telephone support, individual consulting and contract programming to help you develop and install your applications.

Call us today at 800-323-3629 for more information; in Canada or Illinois, dial 312-303-6300. Or write **mdbs**, P.O. Box 248, Lafayette, IN 47902, TELEX 209147 ISE UR.

MDBS III®
ABSOLUTE POWER

CIRCLE NO. 211 ON READER SERVICE CARD

mdbs®

ever, Bezier curves also can be S-shaped, having one inflection point between the two endpoints; this is done by placing the control points on opposite sides of the curve. A Bezier even can be constructed with a loop between the two endpoints (see photo 1).

The Bezier primitive includes the expected options: Arrow, Template, Marker to mark the locations of the end points and the control points, Line to display temporary guidelines as an aid to creating continuous curves, the First and Second options to allow the control points to be positioned independently, and Place, which fixes the control points without accepting the curve.

The Line option is useful because it ensures that two successive Beziers are

VersaCAD includes some drawing primitives that are omitted from other CAD programs: rectangles, polygons, and ellipses.

continuous—that is, tangent to the same straight line. This is accomplished by placing the first control point for one segment on the tangent line placed by the previous segment. Bezier curves can be fitted through a series of data points, but with considerably more control than that offered by a simple curve-fitting command.

Points can be placed in VersaCAD drawings with the ADD Point command. They can be assigned a line width and drawn as arrowheads, marker crosses, or templates. A Rotate option allows points drawn as arrowheads to be oriented as desired.

VersaCAD's tenth primitive, Text, allows a line or block (paragraph) of text to be entered and then positioned. The simplest way to position text is to drag it to the desired location with the pointing device. The cursor control keys for moving text are not the arrow keys, nor even the standard WordStar-diamond keys, but instead, the Ctrl key with *I*, *J*, *K*, and *L* for up, left, down, and right, respectively.

GRIDS, SNAPS, GUIDES

VersaCAD provides the two basic drawing aids, grids and snap. A grid is an array of dots consisting of vertical and horizontal lines with larger dots at the intersections. A grid is defined from the UNITS command menu with the Grids option. The program prompts for *X* and *Y* spacing of the grid lines, which can be set independently. The spacing between dots on the lines, which provides subdivisions between the grid lines, is set when the grid is defined. Grids can be toggled on and off from the SWITCHES menu.

Snap modes include increment snap, grid snap, object snap, and snap disabled. These modes are selected with the F2 function key; the snap increment is set with the UNITS Increment command. F6 (update) can change the increment while another command is in process.

VersaCAD provides another drawing aid in the form of object tracking, which corresponds to the drag mode of some CAD systems. When the object-tracking mode is on, the program continuously redraws objects as they are moved, easing the task of positioning an object. Complex objects can take several seconds to redraw, making the movement jerky. In such cases, object tracking is best turned off.

VersaCAD's CONSTRUCT menu provides several options that can be classified as drawing aids, including guidelines, which perform a function similar to that of grid lines. They are used in precisely the same manner as are the construction or projection lines in manual drafting. Like grids, guidelines do not become a permanent part of the drawing and, in fact, disappear when the screen is redrawn. Unlike grid lines, however, guidelines cannot be turned off and on arbitrarily. They appear on the screen as lines of dots.

Other drawing aids included on the CONSTRUCT menu are normal, parallel, and tangent lines. The Normal and Parallel options each display a submenu that is similar to the menu presented for adding lines. Normal and parallel lines can be constructed as arrows, templates, or with markers displayed. In addition, snap options are available to lock the line to the reference line and to release the cursor.

The Tangent option allows the construction of lines tangent to circles, arcs, and ellipses at one or both ends. If a line is tangent at one end, it can be constructed as normal or parallel to another reference line.

A number of drawing aids can be controlled from the SWITCHES menu, which includes the selections: Marker, Template, Grids, Hatch, Symbols, Outline, Reverse, Upright, All Levels, Levels, Sketch, Quit, and ?Help. All selections

Gateway to anyware.

Thanks to our local and remote communications solutions, your PCs now have total access to the world around them. They can talk to other PCs. Share hardware and software. And take advantage of all the major networking standards. From NetWare to NETBIOS to SNA to X.25.

G/NET™ is our high-performance, low-cost Local Area Network for IBM PCs, XTs, ATs and compatibles. Incorporating the Novell NetWare/G family of LAN operating systems, G/NET offers electronic mail, print spooling, multiple file and communications servers, and access to over 3,000 LAN applications.

And with our G/NETBIOS™ emulation option, G/NET can even run programs developed for IBM's PC Network or Token Ring. This gives you the best of both worlds: IBM LAN software compatibility coupled with the performance advantages of G/NET hardware.

Our G/SNAnet™ gateway provides SNA mainframe links for networked PCs. Up to 32 concurrent sessions are supported through one shared, board-level controller. Each PC can switch between 3270 emulation, 3770/RJE emulation, and local PC-DOS or LAN applications.

Our G/X25net™ LAN gateway lets multiple G/NET users access remote LANs, mainframes and dial-up services. Over X.25 trunk lines or public networks such as Telenet and Tymnet. Finally, our G/Async™ and G/Carbon Copy™ links can be used in a variety of async applications.

Think of us as your PC SafetyNet.™ In addition to our cost-effective products, we offer the security of a total solution supplier. With complete service and support. Continued innovation so you won't be left behind with yesterday's technology. And assured performance and reliability. Just ask any one of our more than 50,000 users.

So call today for your local Gateway distributor or dealer. And let your PCs explore our gateway to anyware.

Gateway communications, inc.
Your PC SafetyNet.™

CIRCLE NO. 139 ON READER SERVICE CARD

16782 Red Hill Ave., Irvine, CA 92714. (714) 261-0762.

except the last two control the display of their respective features. For example, a drawing can be created with template lines for use as construction, projection, or match lines. These lines can be dimensioned and snapped to during the process of creating the drawing, but turned off when the drawing is complete. Grids, Hatch, Symbols, and Outline provide similar control over the visibility of grid lines, hatch lines, symbols, and symbol outlines. The All Levels and Levels options control the visibility of objects by levels. All Levels causes all levels to be visible, and Levels allows individual levels to be turned off and on. Sketch controls whether the program, when redrawing the screen, begins with the first or last object added to the drawing.

DRAWING MODIFICATIONS

Changes can be made to drawing objects with the commands provided in the MODIFY menu: Move, Copy, Rotate, Scale, Image, Text, Delete, Undelete, and Properties. Each of these performs a function on the current object, which is normally the last object added to the active drawing work file. When the user selects the MODIFY menu from the main menu, the current object begins to blink if object-tracking mode is on. To modify another object in the drawing window, the commands **<backward** and **>forward** are invoked. Pressing the less than (<) or greater than (>) key or repeatedly selecting the **<backward** or **>forward** commands from the menu will step through the drawing database, causing each object in turn to blink. Alternatively, the Find command can be used to select an object by pointing to it with the screen cursor.

Move, Copy, Rotate, Scale, and Image display submenus of options that perform variations of the basic command. When Move is selected, the current object jumps so that its *handle point* is positioned at the screen cursor location. Each VersaCAD drawing object has a handle point—a point on the drawing that follows the screen cursor during the operation of the MODIFY command options (and some other commands as well). As the cursor is moved, the object follows. If the default handle is not suitable, an alternative can be selected in one of two ways.

The MODIFY menu includes a Handle option that defines a new handle by pointing to it with the cursor. This allows a handle point to be defined anywhere in the current window, not just at a feature of the object. MODIFY commands, such as Move, that depend on the handle point for manipulation of the object also provide a Handle option; it cycles the current handle through several predefined locations. If the MODIFY Handle option has been used to define a new handle point, the Handle option of the Move command will cycle through that handle point as well as through the predefined features of the object.

Move provides *X*- and *Y*-axis lock options to ensure movement in the *X* or *Y* direction, an Original option to return the object temporarily or permanently to its original location, Free to release *X* or *Y* locks, and Swap to reverse the direction of certain objects, such as lines with arrowheads.

The Copy option creates single or multiple copies of the current object. The Handle option is provided and works as it does in the Move command. The Repeat command creates multiple copies arranged in an array—one direction, two directions, or circular.

Rotate is used to change the angular orientation of the current object. When this command is selected, two markers appear on the object—a pivot point and a handle point. The handle point follows the screen cursor, causing the object to rotate around the pivot point. The cursor need not be positioned near the object, because the object rotates so that the handle point remains on a straight line beginning at the pivot point and passing through the cursor. *X*- and *Y*-axis lock options are not provided, but the Zero option aligns the pivot point and the handle point with the *X*-axis.

An object's size can be changed with the Scale option. When Scale is selected, two markers appear: a stationary point and a handle point. The handle point follows the screen cursor and remains on a projection line from the cursor to the *X*-axis. The *Y*-scale follows. The Scale submenu includes a Swap option, which according to the manual changes the direction only of directional objects. In reality, the Swap command interchanges the stationary and handle points. With appropriate use of the MODIFY Handle, Swap Handle, and Swap options, the point about which an object is scaled can be controlled. The alternative is to scale an object using the default stationary point and then moving it into final position.

Objects can be scaled equally along both the *X*- and *Y*-axes or independently in the two directions. The Unproportional option of the Scale command allows the handle point to follow the cursor exactly, scaling the *X* and *Y*

dimensions accordingly. Options on the Unproportional menu, *X*-axis and *Y*-axis, limit scaling to the respective dimensions, regardless of the cursor's position along the other axis.

Although a mirroring command is provided in almost every CAD package, VersaCAD's Image option is noteworthy, because it allows the image axis to be moved and rotated to any position and orientation or to be forced to vertical or horizontal positions. The default operation of the command erases the original, but a Copy option retains the original while making the mirror image.

The Text option of the MODIFY command allows text to be edited, justified, and adjusted. The Edit option functions much the same as the Text command of the ADD menu, providing the ability to type in new text, delete lines or characters, and accept or reject the changes. With the *I*, *J*, *K*, and *L* control keys mentioned above, text can be moved and/or rotated during editing. The left arrow key erases each character as it backspaces through the line of text being edited; the Del key erases the entire line; and the down arrow moves the line to the left. Text can be justified to the right, left, center, top, and bottom with the Justify option. The Adjust option flips rotated text to read correctly from left to right.

The MODIFY Properties option is used to change the properties assigned to an object when it was created. This Properties option generally corresponds to the PROPERTIES command on the main menu. The main PROPERTIES command defines a set of default—or current or global—properties assigned automatically when objects are created. These properties are Line Style, Density, Line Width, Level, Pen, Color, Group Name, and Text (height, width, and spacing). Each can be changed with the MODIFY Properties option. In addition, the option allows changing the rotation angle, adding or removing arrowheads, switching the template feature, and turning markers off and on. The selections on the submenu of the Properties option are used to define proposed properties; the properties are actually changed by explicitly entering the Update selection on the MODIFY menu (not the Update function key, which can be used to change properties while other commands are active).

COMPLEX OBJECTS

VersaCAD provides two facilities for managing complex objects composed of primitive or other complex objects: groups and symbol libraries.

The group facility is similar to blocks in other CAD packages. A *group* is a collection of objects, defined by the user to perform operations on the group members simultaneously. Although an arbitrary number of groups can be defined, only one is active, or current. The GROUP command menu is used to define and manipulate a group. The options on this menu correspond to those under the MODIFY command: Move, Copy, Rotate, Scale, Image, Properties, Delete, and Undelete. In addition, the GROUP menu includes List, Where, Number, Guides, Sketch, Quit, and ?Help. The List option displays a list of *potential* groups—that is, it displays a list of group names in the work file and the number of objects that share each group name.

One of the properties assigned to each object as it is added to the drawing is the current group name, defined with the main menu PROPERTIES command or the Update function key menu. The group name is not the group identifier, but one of the attributes that can be used to build a group.

Each drawing object has a handle point that follows the screen cursor during the operation of the MODIFY command options.

The GROUP command includes options to build a group in several ways. Objects can be added to or removed from the current group by pointing, surrounding them with a "fence," or on the basis of properties. Groups also can be built from objects associated with several group names.

VersaCAD's second facility for managing complex objects is the symbol library. The LIBRARY menu in VersaCAD is designed to facilitate the management of permanent symbol libraries, rather than to the creation of symbols along the way. A VersaCAD library is a collection of up to 100 symbols, referenced by number, in a single file. The symbols are VersaCAD drawings, created and saved to disk in a drawing session.

Libraries are designed to be used with digitizer overlays. The overlay can be created with the Overlay option of the LIBRARY command, which must be entered when the work file is empty.

The option prompts for a library file name and then draws the digitizer overlay on the screen. The overlay can be plotted with the Output Plotter command, using a predefined plot specification called Overlay.

A library file can be updated by essentially the same process used to create it. The library file is selected with the Active command, new symbols are added, and old ones deleted. When the modifications have been made, a new overlay is plotted. Each overlay plot includes the name and date of the file used to plot it; this simplifies the task of ensuring that the overlay is current.

A library file is selected for use in a drawing with the Active command, and the corresponding overlay plot is placed on the digitizer. The digitizer must be calibrated with the Boundary option. Because overlays consist of a 10-by-10 grid, the only requirement is that the lower left and upper right corners be digitized.

Symbols are added to drawings with the Symbol command from the ADD menu. When selected, Symbol prompts the user to select the desired symbol with the digitizer stylus (or cursor). If the program is configured for keyboard input or for a mouse, then the overlay can be used as a reference sheet, and the symbol number entered at the keyboard.

Several options are available when adding symbols. The symbol can be rotated, imaged, and scaled. In addition, the handle point and properties can be changed, and the symbol can be drawn in outline to speed redraws. The symbol option also provides an option to erase previously entered symbols.

Symbols are inserted into a drawing as single objects. When the first instance of a symbol is put into a drawing, the symbol definition is copied from the library file. Subsequent instances of that symbol reference that definition. A reference, or instance, cannot be edited, although it can be scaled, moved, copied, etc.

A symbol reference can be replaced by a copy of the symbol definition using the Explode option of the MODIFY menu. This replaces the symbol reference with its component parts and allows the parts to be edited.

ADDING DATA TO CAD

The basic VersaCAD program is strictly for graphics. An optional module allows nongraphic attributes to be associated with drawing objects. VersaDATA is the VersaCAD Advanced drafting program and a database management system

integrated into a single program. The package is designed to replace the basic version of the program, although both versions can be installed on the same computer in separate subdirectories. Although VersaDATA is not the only microcomputer CAD system capable of attaching nongraphic attributes to drawing objects, the package adds a capability not found in many products.

Some programs require that the attribute-value pair be defined when the symbol is created—that is, separate symbols are necessary for red chairs, blue chairs, etc. Others, including VersaDATA, allow symbols to be assigned attributes and symbol references to be assigned values for each attribute; thus, one symbol definition can accommodate red chairs, blue chairs, etc.

VersaDATA extends the concept of data management concept even further. Most CAD systems with the capability of attaching nongraphic attribute-value pairs to drawing objects stop at the

Essential to VersaDATA's facility to make reports from drawings is its ability to take objects from the database to include in a report.

assignment of these values and some means to export the data in raw form. Typically, the data must be processed by external programs in order to produce useful reports. VersaDATA, however, displays input forms and report forms on text screens that are developed from within VersaDATA, so external programs are not necessary.

When a drawing has been created and a database developed for the drawing, both can be on-line simultaneously. Changes made to the drawing are reflected in the appropriate database report form, and changes made to the database report forms are reflected in the drawing. In a single screen configuration, either the text screen or the graphics screen (with menus) is displayed; with a dual screen configuration, the database forms and the drawing can be displayed simultaneously.

Creating a database of nongraphic attribute-value pairs is straightforward. Objects are drawn without regard to the attributes that will be assigned. Likewise, the database is created without

regard to the drawing objects to which the attributes will be assigned. The two operations are independent.

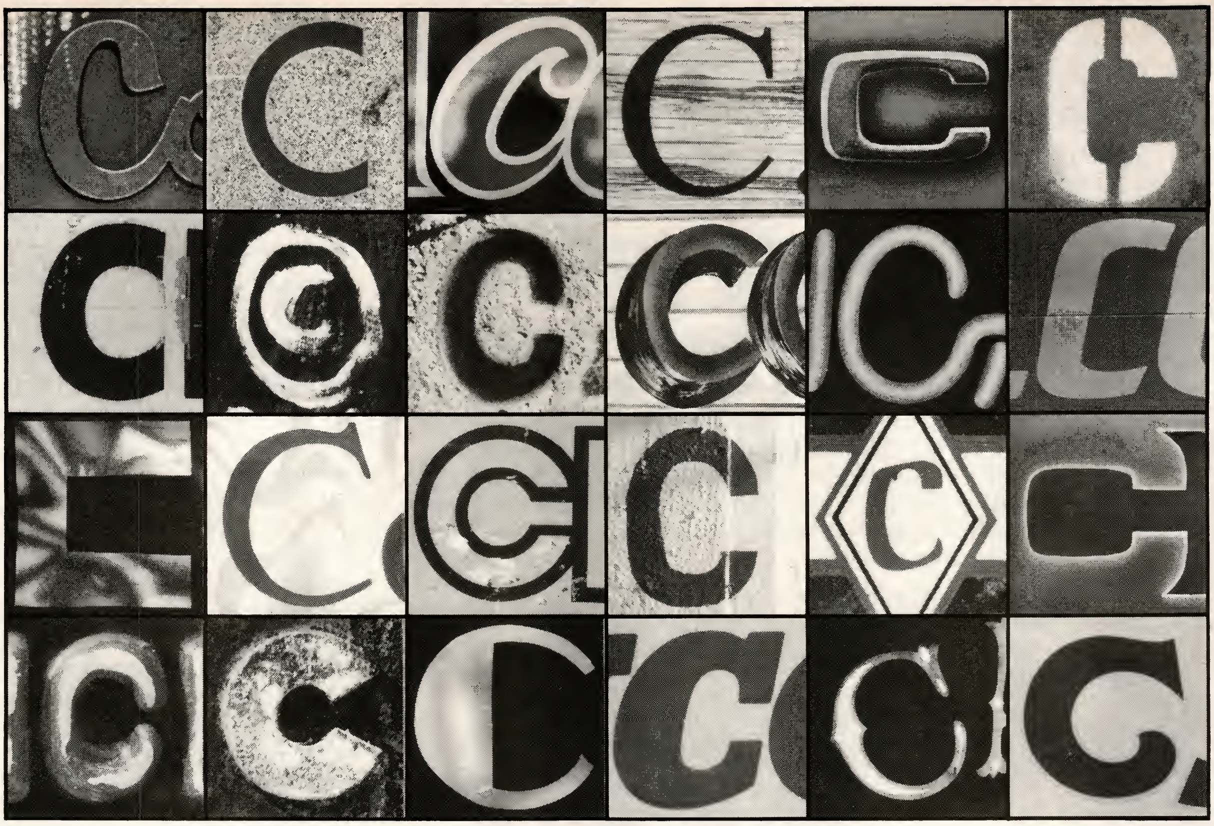
A VersaDATA drawing is created using the menu structure of VersaCAD. The only difference is that VersaDATA has an added main menu selection, called RELATE, that provides access to the database management facility. The RELATE selection displays a menu with the options Create, Forms, Infodata, Masks, Reports, Data, Global ops, Switches, Sketch, Quit, and ?Help.

The process of creating a database begins with the Create option, which prompts for the number of records expected and their length. The program displays a default of 100 for the number of records and a notice that the maximum number of temporary records is 500. The program also presents a default record size of 50 characters. These numbers can be increased to as many as 32,767 records and 4,096 characters per record. By database management standards, even these maximums are modest, but should prove adequate for drawing-related database management.

When the database has been created, the input forms must be generated with the Forms selection from the RELATE menu. Input forms are used to enter the nongraphic data, just as with a database management system. The Forms option displays a menu with the selections Create, Edit, Delete, List, Quit, and ?Help. Each of these options displays yet another menu.

Nongraphic data are associated with the graphic objects in a drawing using the Infodata selection of the RELATE menu. Infodata assigns values to attributes. Attributes are associated with objects when the input forms are created. Creating a form does not bind attributes to objects as is done in some CAD programs. An input form can be created for associating data with chair symbols, but not every chair symbol needs to have data entered. Furthermore, if a drawing contains symbols for tables and symbols for chairs, and input forms for each, the program will not complain if the table form is used to associate data with a chair symbol.

When the user selects the Infodata option, the current object begins to blink, just as with the MODIFY command. The Infodata menu includes the selections Form, Edit, Delete, Find, <backward, >forward, Sketch, Quit, and ?Help. The Form option is used to select the form that will be used to enter data, and the Edit command displays the form and accepts the data. Delete removes data from the form associated

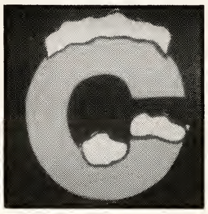


NEW! FROM BLAISE COMPUTING

C Tools Plus™

For The Programmer Whose Alphabet Begins & Ends With "C"

Today's programmers need more than yesterday's tools. Requirements such as removable windows and "sidekickable" pop-up utilities are changing the face of program design. You need to filter interrupts so that other resident programs still work. You need the ability to switch between multiple display pages and monitors. Today's technical demands are almost endless, but C TOOLS PLUS gives you what you need.



SOLID LIBRARY SUPPORT

Blaise Computing offers you solid library support that can meet all your demands and more. C TOOLS PLUS embodies the full spectrum of general-purpose utility functions that are critical to today's applications.

Here's just part of the PLUS in C TOOLS PLUS:

◆ **C TOOLS and C TOOLS 2** compatibility—two packages that receive rave reviews for quality, organization, usability and documentation.

◆ **FULL SOURCE CODE**

◆ **WINDOWS** that are stackable, removable, that support word wrap and that can accept user input.

◆ **INTERRUPT SERVICE ROUTINE** support for truly flexible, robust and polite resident applications.

◆ **MULTIPLE** monitor and display support, including EGA 43-line mode.

◆ **FAST DIRECT VIDEO ACCESS** for efficiency that will not constrain good program design.

◆ **DOCUMENTATION, TECHNICAL SUPPORT** and attention to detail that have distinguished Blaise Computing products over the years.

*C TOOLS PLUS supports the Microsoft (and IBM) 3.00 and Lattice 3.00 C compilers and is just **$175.00.***

Also Available Are:
C VIEW MANAGER—*A kit for building data entry screens and menus. Begin by designing on-screen what the operator will see; call upon our library functions from your program to display the screens and retrieve the data. Just $275, including all library source code.*
C ASYNCH MANAGER — *provides the crucial core of hardware interrupt support needed to build applications that communicate. It also includes the "XMODEM" file-transfer protocol and support for Hayes-compatible modems. All source code is included for $175.*
C TOOLS & C TOOLS 2—*an indispensable combination still available at a low price of $175, including all source code. See review in* PC Tech Journal, 6/85.

BLAISE COMPUTING INC.
2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441
ORDER TOLL-FREE 800-227-8087!
CA residents call (415) 540-5441

YES, send me the PLUS I need! Enclosed is $______ for C TOOLS PLUS. (CA residents add 6½% Sales Tax. All domestic orders add $10.00 for Federal Express shipping.)
Name: ______ Phone: (___) ______
Shipping Address: ______
City: ______ State: ___ Zip: ______
VISA or MC #: ______ Exp. Date: ______

CIRCLE NO. 114 ON READER SERVICE CARD

with the current object, and the remaining options operate the same as the corresponding options of MODIFY.

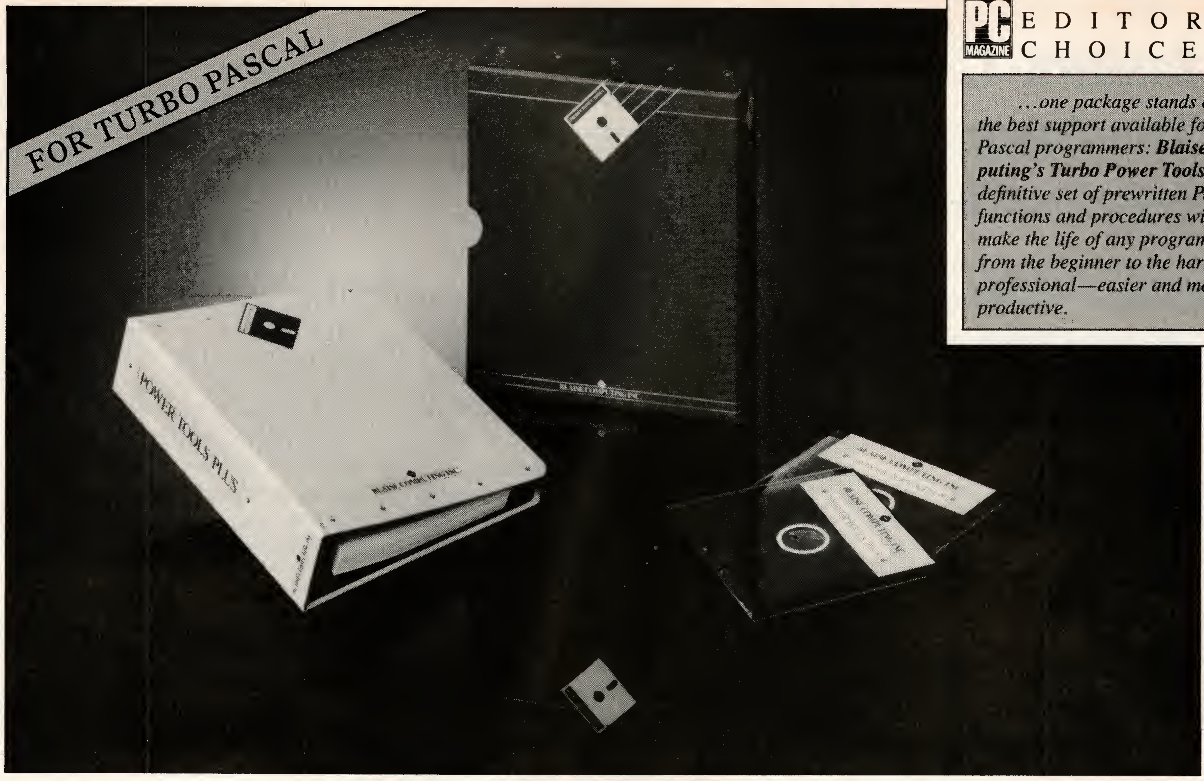
With the Edit option, the screen changes to a text display of the selected input form. The first line of the display contains the drawing name, the form number, the form name, and a *masterkey* used to index the drawing object to the database form. The body of the form contains the fields defined with the Form Create command. The last two lines contain fields that describe the graphic object: group, level, color, pen, style, library name and number, and *X* and *Y* coordinates.

Entering the masterkey associates attributes with the drawing object in that a copy of the form is then associated with the object via VersaDATA's indexing scheme. Completing the body of the form binds attribute-value pairs to the object. When the form has been associated with the object, the graphic attributes are available in the form.

Essential to VersaDATA's facility for producing reports from drawings is the ability to select objects from the database for inclusion in a report. This selection process is accomplished with the use of masks, defined with the Masks menu. A *mask* is a set of logical expressions used to search for objects that meet a set of conditions—for example, price > 1,000 (and) size = 60. A mask is created from a copy of a Form, with the Create option of the Masks menu. The blank form appears on the screen, just as it does when the Edit option of the Infodata menu is selected. However, instead of data, the user is expected to enter an expression beginning with one of the following logical operators: [(begins with);] (ends with); = (equal to); # (not equal to); < (less than); and > (greater than).

The expression entered by the user, together with the field name, forms a complete logical condition, and the set of logical conditions forms the mask. Not each field name needs to be followed by a logical expression, and several masks can be created from each form. In addition, masks can be linked with the logical operators A(nd) and O(r). However, a field can contain only a single expression; linking masks requires completing multiple screens.

Logical conditions are not limited to nongraphic fields. The graphic properties listed at the bottom of the form also can be used as the basis of logical conditions; for example, a mask can be used to select all chair symbols from the Acme Chair Company in the upper left quadrant of the drawing with prices



PC MAGAZINE EDITOR'S CHOICE

*...one package stands out as the best support available for Turbo Pascal programmers: **Blaise Computing's Turbo Power Tools.** This definitive set of prewritten Pascal functions and procedures will make the life of any programmer—from the beginner to the hard-core professional—easier and more productive.*

ANOTHER PLUS FROM BLAISE COMPUTING

Power Tools Plus™
Window Routines.
Memory Resident Routines.
Routinely.

The best just got better! Turbo POWER TOOLS, acclaimed as the best programmer support package for Turbo Pascal, now has even more functions, more detailed documentation and more sample programs.

NO SECRETS

Turbo POWER TOOLS PLUS is crafted so that the source is efficient, readable and easy to modify. We don't keep secrets! We tell you exactly how windows are managed, how interrupt service routines can be written in Turbo Pascal, and how to write memory resident programs that can even access the disk. Maybe you've heard of some undocumented DOS features that resident programs use to weave their magic. Turbo POWER TOOLS PLUS documents these features and lets you make your own magic!

Here's just part of the PLUS in Turbo POWER TOOLS PLUS:

- **WINDOWS** that are stackable, removable, with optional borders and a cursor memory.
- **FAST DIRECT VIDEO ACCESS** for efficiency.
- **SCREEN HANDLING** including multiple monitor and EGA 43-line support.
- **POP-UP MENUS** which are flexible, efficient and easy to use, giving your applications that polished look.
- **INTERRUPT SERVICE ROUTINES** that can be written in Turbo Pascal without the need for assembly language or inline code.
- **INTERVENTION CODE** lets you develop memory resident applications that can take full advantage of DOS capabilities. With simple procedure calls, you can "schedule" a Turbo Pascal procedure to execute either when a "hot key" is pressed, or at a specified time.
- **PROGRAM CONTROL ROUTINES** allow you to run other programs from Turbo Pascal, and even execute DOS commands.
- **MEMORY MANAGEMENT** allows you to monitor, allocate and free DOS-controlled memory.
- **DIRECTORY AND FILE HANDLING** support to let you take advantage of the newer features of DOS including networking.
- **STRING** procedures allowing powerful translation and conversion capabilities.
- **FULL SOURCE CODE** for all included routines, sample programs and utilities.
- **DOCUMENTATION, TECHNICAL SUPPORT** and attention to detail that have distinguished Blaise Computing over the years.

Turbo POWER TOOLS PLUS** supports Turbo Pascal Version 2.0 and later and is just **$99.95.

*Another quality product from Blaise Computing: **Turbo ASYNCH PLUS™***

A new package which provides the crucial core of hardware interrupt support needed to build applications that communicate. ASYNCH PLUS offers simultaneous buffered input and output to both COM ports at speeds up to 9600 baud. The XON/XOFF protocol is supported. Now it also includes the "XMODEM" file-transfer protocol and support for Hayes compatible modems.

The underlying functions of Turbo ASYNCH PLUS are carefully crafted in assembler for efficiency and drive the UART and programmable interrupt controller chips directly. These functions, installed as a runtime resident system, require just 3.2K bytes. The high level functions are all written in Turbo Pascal in the same style and format as Turbo POWER TOOLS PLUS. All source code is included for just $99.95.

BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

ORDER TOLL-FREE 800-227-8087

CA residents call (415) 540-5441

YES, send me the PLUS I need! Enclosed is $______ for ☐ Turbo POWER TOOLS PLUS ☐ Turbo ASYNCH PLUS ☐ OTHER ______. (CA residents add 6½% Sales Tax. All domestic orders add $10.00 for Federal Express shipping.)

Name: ______ Phone: (___) ______

Shipping Address: ______

City: ______ State: ______ Zip: ______

VISA or MC #: ______ Exp. Date: ______

CIRCLE NO. 102 ON READER SERVICE CARD

greater than $150 and color beginning with red. The number of masks could be large in an extensive drawing library, so the List option of the Masks menu includes a Catalog option to view a directory of existing masks and a Display option to display masks on the screen or print them on the printer.

VersaDATA's reports are typical of the basic report formats contained in most database management systems. The Reports menu presents options to create, save, and delete report formats. Selected fields of a number of database records are printed in a free-form layout. The report layout is created by selecting the desired fields from a copy of the form and editing the page layout from a line format description of the report (see photo 2); the screen display is similar to the line format that is used to edit input forms.

Reports are produced from the Reports option of the RELATE menu. They can be directed to the screen, printer, or to a disk file, with or without formatting information.

Forms, masks, and reports are not linked to a specific drawing. This allows the creation of an application library that is able to produce standard reports from a number of drawings. For example, a facilities management system could be built using the drawings of the building and a library of forms, reports, and masks. The drawing could contain symbols representing furnishings, equipment, and people; the symbols could be relocated graphically as the items or people were moved from one space to another. Following the relocation of an office or department, reports could be generated to reflect the current assignment of assets, building directories, etc. Other applications of VersaDATA's report generating facility include the generation of cost estimates, bills of material, and parts lists.

SPECIAL REPORTS

Whereas VersaDATA can be used to produce a variety of reports, VersaLIST is more specialized in that it produces certain kinds of reports: bills of material and cost estimates. VersaLIST is a stand-alone program that processes drawing files, while VersaDATA is an on-line database management system integrated with VersaCAD. VersaLIST also differs from VersaDATA in the way attribute-value pairs are handled. VersaDATA allows attributes to be associated with symbol definitions and attribute values to be assigned to symbol references; VersaLIST requires that attribute values be assigned to symbol definitions.

VersaLIST is not a graphics program but a utility that processes drawing files. The program is menu driven, and the menus are placed in the same location as menus in the drafting programs, easing the learning process.

The main menu selections are EDITOR, LOAD, COUNT, REPORT, FILER, and EXIT. The EDITOR selection creates the various tables VersaLIST uses to create reports: Match, Description, Labor, and Count. The Description and Labor tables are optional.

A bill of material or cost estimate is created in four basic steps. The first step is to create a Match table using the EDITOR-Match-Add command string from the main menu. The Match table is

Whereas VersaDATA can produce a variety of reports, VersaLIST is more specialized in that it generates certain kinds of reports.

used to associate a symbol in a specific library with a key, which consists of code and size fields. The titles of these fields do not indicate a required data type. The code for an item can be a name, serial number, part number, or any other alphanumeric string. Similarly, the size field can contain an actual size, color, model number, or other alphanumeric string. Together these two fields produce a unique key, even though the entry in either field may be duplicated. Therefore, the code field can have several entries for chairs, as long as each one contains a unique entry in the size field.

However, a specific symbol—that is, a symbol from a specific library—can be bound only to a single code-size combination. Unlike the arrangement provided with VersaDATA, VersaLIST requires that separate symbols be created for red, blue, and green chairs.

The Match table can be accompanied by Description and Labor tables, which produce more detailed reports. Creation of all three tables is straightforward. The program presents an input screen that allows full-screen editing with the arrow keys.

The second step in the preparation of a report is generating the Count table with the COUNT command from the main menu. The program requests

the name of the Match table to be used, the name to be assigned to the Count table, and the name of the drawing to process. The program scans the drawing file and counts the instances of each symbol that appears in the Match table. Several drawings can be processed in a single session. When one is finished, the program requests the name of the next file to process.

Note that a symbol can be bound to more than one attribute-value pair by the use of different Match table files. However, at the time the drawing is being processed by the COUNT function, only a single binding is in effect. Thus, a drawing containing symbols for light fixtures can be processed several times using different Match tables to produce bills of material for the fixtures themselves, wiring connections, junction boxes, or any other item that can be directly related to the symbol.

After the Count table is created, it can be used as is or processed by commands available in EDITOR menu. Specifically, entries can be edited, deleted, duplicated, and sorted.

When the Count table is satisfactory, a report specification must be created using the REPORT/SPECS command from the main menu. The specification lists the names of the tables to be used, printer configuration, headings, and the fields to be used in the report. The fields are predefined; the user can specify the order in which the fields are to appear in the report. The fields themselves and the headings cannot be changed.

A report specification is not tied to specific drawings or Match tables. It can be used with a number of different drawings and Match tables in order to produce standard reports.

The final step in producing a report with VersaLIST is to invoke the command REPORT/GENERATE to produce a report file on disk, which then can be printed with the Print command of the REPORT menu. As an alternative to printing the report, the file can be processed by the supplied TWG program. This program converts the report file to a drawing exchange file in ASCII text form, called a TWGES file.

The TWGES file format is intended to transfer VersaCAD drawings between different versions of the program and to allow external processing. T & W provides an extra-cost option, the IGES (initial graphics exchange specification) module, to convert the TWGES file to an IGES file that can be transferred to any system that supports the IGES format or any system for which an IGES translator program is available.

Processing a report file with the TWG program produces a file containing the report and additional information required by the TWGES specification to describe a drawing in text form. A TWGES file can be processed by the VersaLINK program to convert it to a drawing file readable by VersaCAD.

Thus, a drawing can be processed to count symbol instances, and to produce tabular reports showing the quantities and nongraphic attributes of those instances. The tabular reports can be converted to drawing database format and merged into the original drawing or used as a separate drawing. This streamlined method of producing bills of material and schedules automatically, without the need to purchase software from other vendors, is a sign of a well-designed CAD system.

In addition to the optional VersaLIST and VersaCAD modules, T & W Systems offers the 3-D Enhancement system to meet the need for three-dimensional modeling. The 3-D system can be used in conjunction with either VersaCAD or VersaDATA. This optional program is reminiscent of the 3-D modeling systems written for eight-bit microcomputers several years ago. A 3-D model is constructed by creating a table of coordinates, rather than in an interactive manner. VersaCAD 3-D falls somewhat short of an interactive 3-D modeling system and is severely limited in the size of scene that can be created.

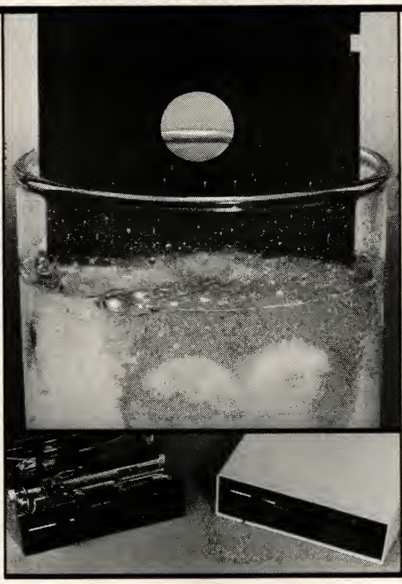
Perhaps a better alternative at this time is to purchase a separate 3-D modeling program that is able to communicate with VersaCAD via a drawing exchange file. Design Board Professional from MegaCADD is one such product (it was reviewed in "Three-dimensional Modeling," Victor E. Wright, February 1986, p. 88).

VERSACAD COMPANION

The latest version of VersaCAD Advanced, introduced shortly before this review, does not replace the previous version, but is currently a companion program—more versatile, powerful, and expensive. The initial release of VersaCAD 5.0 was not accompanied by new releases of VersaLIST, VersaDATA, and the 3-D modeling program. These updated options were scheduled for release late this summer.

GET PC/XT HEADACHE RELIEF.

Take one WELTEC floppy drive and upgrade your PC or XT to AT capacity - without changing your controller card.



Is your mix of IBM PC's and XT's giving you an MIS headache? Get rid of it fast with our versatile 5¼" half-height flexible disk drive.

- 1.2 Mb formatted capacity
- No additional controller card needed
- Brushless direct-drive motor with 30,000 MTBF
- Full interchange of diskettes between IBM AT and PC or XT as well as most IBM compatible systems
- Internal or external configurations
- ANSI and industry interface compatibility

Get relief now!
Call 714/250-1959

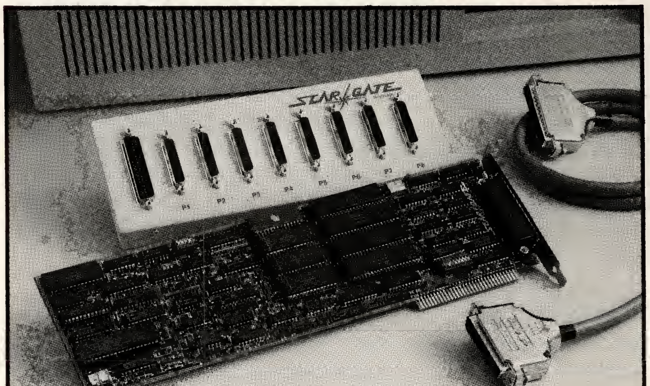
VAD and VAR inquiries invited

WELTEC
digital, inc.

17875 Sky Park North, Suite P, Irvine, CA 92714

IBM™ PC, XT and AT are trademarks of International Business Machines Corp.

CIRCLE NO. 243 ON READER SERVICE CARD



Advanced Communication Link™ Lets Your PC Use All The Brains It's Got

New "smart" 8-channel serial card unloads your IBM PC — or compatible — to give you more computing performance.

Using its own on-board CPU and an array of supporting components, the new ACL card relieves PCs of communication chores with mainframes, printers, modems, plotters, and remote terminals. The PC is thus free to do the "thinking" you demand of it.

In addition, the ACL Card expands your capabilities, too, with:

- Full modem "handshake" control on all 8 channels
- Flexible address decoding
- Programmable interrupts
- Expandable memory via software
- Supplied device driver
- Ability to download custom programs
- Optional data security module

All this takes up only *one* I/O space on your PC — and costs about the same as a "dumb" serial card.

Call for free literature that explains what your PC has been too busy to tell you.

STAR GATE TECHNOLOGIES INC.

33800 Curtis Blvd., Eastlake, OH 44094
(216) 951-5922

CIRCLE NO. 177 ON READER SERVICE CARD

FREE!

DATA ACQUISITION • GRAPHICS • DATA ANALYSIS PRODUCT CATALOGS FOR YOUR IBM or Apple PC

Interactive Microware offers a complete, low cost family of high quality hardware and software for **Chromatography • Process Control • Spectrophotometry • Temperature Monitoring • Electrochemical Analysis • Atomic Absorption • Instrument Monitoring • Data Plotting • Curve Fitting • IR and UV/VIS Spectroscopy • Computer Assisted Design** and Much More!

Thousands of clients worldwide use IMI products for Science, Engineering and Business Applications.

CALL (814) 238-8294 FOR IMMEDIATE ACTION!

featuring

COMPLETE APPLE II WORKSTATIONS STARTING AT $ 2220

COMPLETE IBM PC WORKSTATIONS STARTING AT $ 3595

IBM is a registered trademark of International Business Machines Corp; Apple is a registered trademark of Apple Computer, Inc.

IBM 48 PAGES 94 PRODUCTS **APPLE 64 PAGES 264 PRODUCTS**

imi™
INTERACTIVE MICROWARE, INC.
POB 139, Dept. 237, State College, PA 16804
Phone: (814) 238-8294 • Telex 705250

CIRCLE READER SERVICE NO. 209 FOR IBM, NO. 208 FOR APPLE

On the surface, VersaCAD 5.0 resembles version 4.0 very closely. This point is important for users who will upgrade to the more powerful version, because the need to learn new commands is minimal. For example, the main menu of VersaCAD 5.0 differs from that of version 4.0 only by the addition of a DIMENSION command.

Symbols and libraries are still a part of VersaCAD 5.0, but the maximum size of a library has been raised from 100 symbols to 1,000. Precision also has been increased to 16 from the 11 decimal digits of version 4.0. With the increase in precision is an increase in the range of numbers. The minimum resolution provided is 10^{-307} and the maximum is $\pm 10^{307}$.

The number of supported graphics devices has been increased from 12 display adapters to 20, from 17 input devices to 21, and from 12 output devices to 14. A new feature is the support of expanded memory specification (EMS) cards to speed panning and zooming.

Use of the function keys has been expanded from 11 functions to 20, requiring the use of the Ctrl key in conjunction with the function keys. The keys previously supported—F1 through F8—have not been redefined, and the Ctrl-F*x* keys generally complement the previous functions.

Function keys F9 and F10 are used with an entirely new feature, keystroke/data entry macros. The F9 key controls the definition of a macro. Pressing F9 once begins recording a macro in memory, and pressing it a second time ends the recording process. Ctrl-F9 displays a menu of commands used to save macros to disk and to execute a macro previously saved to disk. The F10 key executes a macro, and Ctrl-F10 executes the macro currently in memory—the last one recorded with F9.

CUSTOM CAD

The most significant improvements in VersaCAD 5.0 are those features related to customizing the program for specific applications or simply to suit the user's preferences. The facility for customizing VersaCAD 5.0 is the CAD Programming Language (CPL).

CPL is a structured programming language that bears more than a passing resemblance to C. In effect, VersaCAD 5.0 is a CPL interpreter because CPL statements can be entered in response to most prompts.

The language is clearly complete. The general description of CPL and the reference section is more than one-fourth-inch thick in the documentation. The manual concentrates on the construction of macros—sequences of commands assigned to a single command name—but the language offers considerably more power than that.

CPL includes arithmetic, relational, bitwise, and logical operators. Statements are terminated by a semicolon or ending brace. Braces are used to delimit blocks, which can be nested. Comments are delimited with matching /*...*/. Assignment is accomplished with the equal sign. Two constructs are provided: IF...THEN statements, which can be nested 15 deep, and WHILE...DO, which can be nested 10 deep.

CPL provides four data types: real numbers, integers, strings, and characters. Reals, integers, and characters can be mixed in expressions, and the program performs little type checking. The language includes 138 functions. The CPL interpreter does not provide for procedures or user-defined functions within programs. According to the description that is presented in Backus-Naur Form in the documentation, a program is a statement list only. However, macros—which are actually CPL programs—can be grouped together, thereby providing a similar effect.

Macros can be defined without CPL by using the function keys, as mentioned above, or with CPL by writing a command file (disk file of CPL statements) with a text editor.

Another CPL application is preparing custom digitizer overlays and key assignments. Several macro definitions can be placed in a command file. The command file can be associated with a specific area on a digitizer overlay via a tablet overlay file. The program includes the commands to create the overlay drawing in the most suitable format. Overlays can contain areas for pointing to screen menu items, symbol libraries, macros, and graphics pointing.

VersaCAD's CPL can be used to customize screen menus, prompts, and messages. The command text file used in making a custom menu must be converted to a compressed format that VersaCAD reads via its CRUNCH program. CRUNCH also can "decompress" a file so that it can be edited.

In addition to customizing the menus and message files, VersaCAD 5.0 allows the user to create custom text fonts. The process is similar to that used in version 4.0.

PRODUCING HARD COPY

VersaCAD provides two forms of plotted output: plots to a pen plotter and screen dumps to a graphics printer.

Both are produced from the OUTPUT selection of the main menu, which includes Plotter, Printer, Specs, Create Spl, Sketch, Quit, and ?Help options.

The screen dump is initiated with either the Printer option or the Ctrl-F8 key combination. It is of limited use except for interim check plots. The printer plot includes not only the drawing as currently displayed, but the menu area, prompt area, and status line. If the system is configured with a dual screen display, the plot includes only the information that is displayed on the graphics monitor. Plots are printed on the paper sideways, which makes the most of 8½-by-11-inch paper but wastes space on a wide carriage printer.

Plotting to a pen plotter is more involved than producing printer plots. Before a pen plot can be produced, a plot specification must be defined. This is list of parameters that determine the portion of the drawing to be plotted, the scale of the plot, and the position and boundaries of the plot on the paper. Plot specifications are defined from the Specs option, which includes the further selections Window, Boundary, Factor, Maximum, Get, Save, Delete, List, Quit, and ?Help.

Choosing Window determines the area of the drawing to be plotted. When selected, the command causes the entire window to blink and displays a list of options. The plot window can be scaled and/or moved. The default shape is that of the graphics window, but an Unproportional option allows the aspect ratio to be changed.

The Boundary option determines the position of the plot on the paper and the size of the margins. If possible, the Maximum option displays the boundaries for the largest plot that the plotter is capable of producing. The manual states that Hewlett-Packard plotters are able to return this information, and the command also works with the Houston Instrument DMP-52 MP plotter that was used in the review.

The Factor option sets the scale of the plot—the ratio of plotted units to drawing units. When selected, the option presents a scale factor of 1 as the default and asks for plot units—dimensions on the paper—and drawing units.

Get, Save, Delete, and List are used to manage a library of plot specifications within the work file. Specifications can be saved and retrieved by name, which can be seven characters long. If the drawing is saved (with the FILER Save command), the plot specifications currently defined in the work file are saved with it. Used in conjunction with saved windows, the specifications can speed the process of plotting details and revised drawings.

VersaCAD provides a facility to plot drawings from dedicated plotting stations without installing VersaCAD. The Create Spl option from the OUTPUT menu creates a plotter spool file of plotter commands. This ASCII text file consists of commands appropriate to the plotter selected during configuration. The spool file is plotted with a 36-line BASIC program, SPOOL.BAS, furnished with VersaCAD. The speed of the plotting process is controlled by the mechanical limitations of the plotter,

VersaCAD 5.0 is the ideal product for the user who needs to customize or to add design extensions to produce a CADD system.

not of the plotting program, so no penalty is paid for plotting with a program running under the BASIC interpreter usually available on a PC.

The computer that is used to run the SPOOL program does not have to be able to run VersaCAD. SPOOL uses no graphics and requires only enough memory to run BASIC, a serial port to connect to the plotter, and a disk drive that will accommodate the spool file. The concept of off-line plotting from a minimally configured PC is important, because CAD programs demand increasingly powerful microcomputers that are better used as graphics workstations than as plotter hosts.

WORTHY CONTENDER

Judging the performance of a CAD system is largely a subjective process. No two programs have the same command structure, store drawing objects in the same manner, or even perform exactly the same tasks. A package may improve productivity in one discipline by a considerably greater margin than in another, simply because it fits that particular discipline better.

Program speed and ease of use can be of varying importance in assessing a program's performance. Most CAD users soon forget the novelty of creating CAD drawings on a desktop computer and begin to notice any delay in command completion. Perhaps the two

functions in which a delay is most obvious are generating arrays or hatch patterns and regenerating the display to zoom in or out. Testing for this review showed that the speed of VersaCAD was comparable to the speed of one of the leading microcomputer-based CAD programs, AutoCAD.

In the area of ease of use, VersaCAD is on a par with other programs. The single letter commands make command entry fast in most cases. However, digitizer overlays support only symbol entry, and some commands prevent the screen cursor from leaving the graphics window. As a result, a combination of pointing device and keyboard entries is necessary. If a digitizer with a stylus is used, the process can be awkward, because the stylus must be laid down and picked up, or all keyboard entries must be made with the same hand.

VersaCAD's facilities for three-dimensional modeling are somewhat limited, although T & W Systems apparently is working toward an improved 3-D system. For now, a companion program such as MegaCADD's Design Board Professional is a better solution.

The two VersaCAD Advanced products, 4.0 and 5.0, are suitable for production use. Version 4.0 produces high-quality drawings and can generate reports from those drawings automatically. This version could be the basis of a CAD-based management system—for example, a facilities management or space inventory management system.

VersaCAD 5.0 is the ideal product for the user who needs to customize or to add design extensions to produce a CADD (computer-aided drafting and design) system. This newer version can be customized to form a turnkey, discipline-specific CADD system. The entire VersaCAD Advanced system with its VersaLIST and, especially, its VersaDATA components certainly are worthy contenders in the ever-improving microcomputer CAD arena.

VersaCAD 4.0: $1,495
VersaCAD 5.0: $2,495
VersaDATA: $995
VersaLIST: $495
VersaLINK: $500
VersaCAD 3-D Enhancement: $495
T & W Systems
7327 Prince Drive
Huntington Beach, CA 92647
714/847-9960
CIRCLE 353 ON READER SERVICE CARD

Victor Wright is the manager of process engineering at Luckett & Farley in Louisville, Kentucky. This article is his fifth in a series on microcomputer-based CAD systems.

OFFICE AUTOMATION

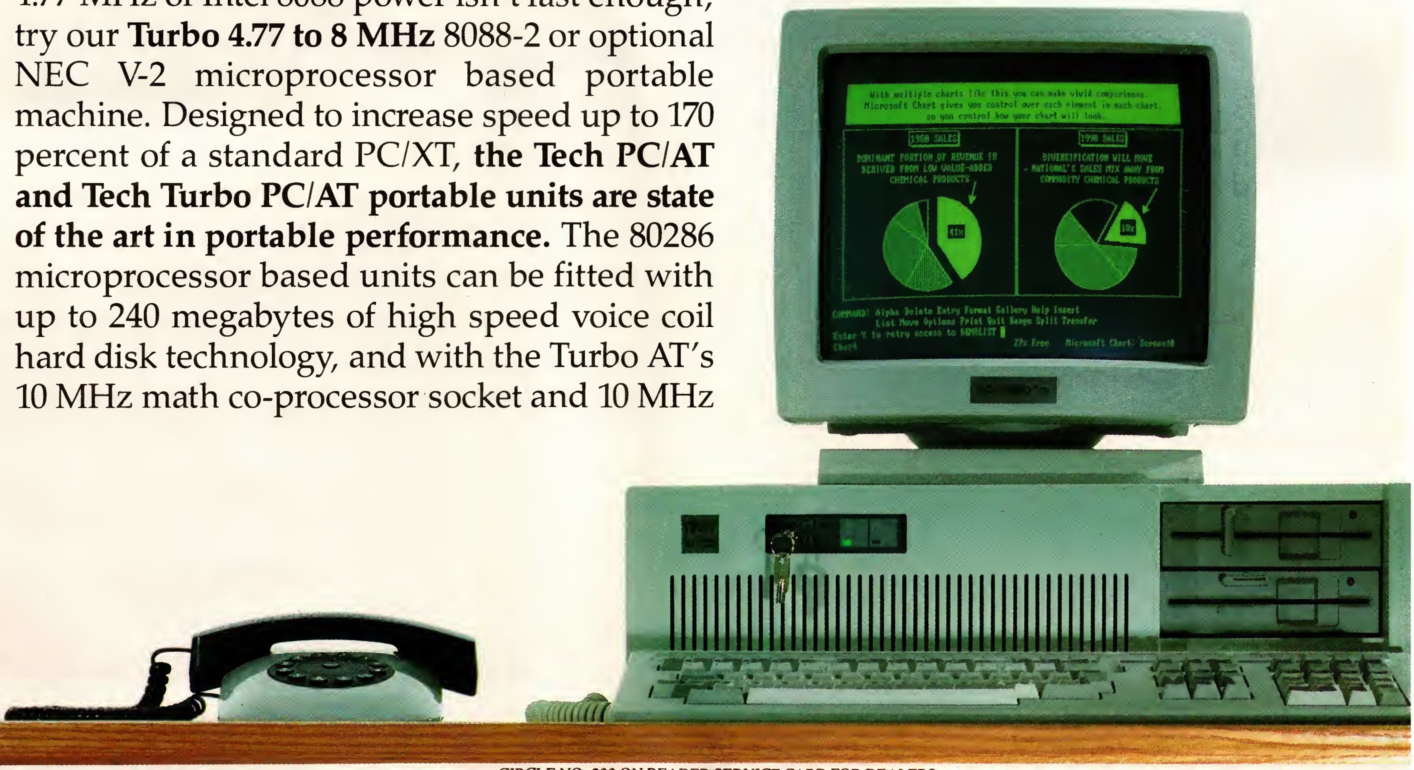
Tech PC, we're out to capture your microcomputer business. No one matches our diverse line of high performance products, or our comprehensive system of support, warranty, and third party maintenance service options. Tech PC is a large Southern California computer design and manufacturing firm with computer research and design specialists, production engineers, technical assistance staff, qualified sales personnel, and a strong management and investment team to provide you with state of the art products, state of the art reliability, and state of the art economy.

PORTABLES

Tech PC Portable Computers come in four different base models to match your computing needs. Our entry level **Tech PC/XT** portable computer is designed to provide a non-obstructive profile, and has a capacity to hold two thirty megabyte hard disks, a high resolution monochrome monitor with a Hercules compatible graphics card to drive it with 800 by 400 lines resolution, and never before heard of 6 expansion slots in a portable computer. If 4.77 MHz of Intel 8088 power isn't fast enough, try our **Turbo 4.77 to 8 MHz** 8088-2 or optional NEC V-2 microprocessor based portable machine. Designed to increase speed up to 170 percent of a standard PC/XT, **the Tech PC/AT and Tech Turbo PC/AT portable units are state of the art in portable performance.** The 80286 microprocessor based units can be fitted with up to 240 megabytes of high speed voice coil hard disk technology, and with the Turbo AT's 10 MHz math co-processor socket and 10 MHz 80286 microprocessor the unit can manipulate large spreadsheets with unparalleled performance.

DESKTOPS

Tech PC Desktop Computers are designed for survival. The **XT's** leave your desktop looking comfortably empty with their slim non-obtrusive profile; The **AT's** give you a powerful presence with a clean lined sturdy chassis. Combined with a 14 inch, 800 by 400 line amber monochrome monitor or a 14 inch, 1280 by 800 black and white monochrome monitor you've got **State of the Art** staring straight back at you. If 8 MHz 80286 power isn't fast enough for you... try the Tech **Turbo PC/AT** with its 10MHz 80286 and 10 MHz 80287 math co-processor socket. The Tech Turbo PC/AT gives you the **strongest, most powerful** 80286 performance available from anyone, anywhere... available with up to 16 megabytes of memory and hard disks up to 1 gigabyte with our new optical storage disk, the Tech Personal



Computer Desktops represent the cutting edge of desktop microcomputer technology.

Tech Personal Computers, Inc. is a full service manufacturer of Micro Computer Products and offers a complete line of Desktop, Portables and Multi-User Computer Systems as well as an accessory line of over one hundred enhancement products. **Tech Personal Computers, Inc.** are all backed by a full one year warranty with additional maintenance coverage and extended maintenance contracts available through Momentum Service Corp. For more information concerning hundreds of MSC Service Centers throughout the United States, contact **Tech Personal Computers, Inc.** at **(714) 754-1170.**

OFFICE ON LOCATION

DESKTOPS

TECH PC/XT DESKTOP .... **$ 529**
Options:
Tech PC/XT with 20MB Hard Disk .... $ 979
Tech PC/XT with 20MB Hard Disk, Monochrome Monitor, Hercules® Compatible Mono/Graphics Card $1179

TECH TURBO PC/XT DESKTOP .... **$ 629**
Options:
Tech Turbo PC/XT Desktop with 20MB Hard Disk $1079
Tech Turbo PC/XT 20MB Hard Disk, Monochrome Monitor, Hercules® Compatible Mono/Graphics Card $1279

TECH PC/AT DESKTOP **$1399**
Options:
Tech PC/AT with 20MB Hard Disk $1799
Tech PC/AT with 20MB Hard Disk, Monochrome Monitor, Hercules® Compatible Mono/Graphics Card $1999

TECH TURBO PC/AT DESKTOP **$1599**
Options:
Tech Turbo PC/AT Desktop with 20MB Hard Disk $1999
Tech Turbo PC/AT with 20MB Hard Disk, Monochrome Monitor, Hercules® Compatible Mono/Graphics Card $2199

PORTABLES

TECH PC/XT PORTABLE .... **$ 899**
Options:
Tech Turbo PC/XT Portable with 20 MB Hard Disk $1349
Tech Turbo PC/XT Portable with 1200 Baud Internal Hayes Compatible Modem and 20 MB Hard Disk $1549

CIRCLE NO. 234 ON READER SERVICE CARD FOR PORTABLE 286

TECH TURBO PC/XT PORTABLE .... **$ 999**
Options:
Tech Turbo PC/XT Portable with 20 MB Hard Disk $1449
Tech Turbo PC/XT Portable with 1200 Baud Internal Hayes Compatible Modem and 20 MB Hard Disk $1649

TECH PC/AT PORTABLE **$1799**
Options:
Tech PC/XT Portable with 20 MB Hard Disk $2199
Tech PC/XT Portable with 1200 Baud Internal Hayes Compatible Modem and 20 MB Hard Disk $2399

TECH TURBO PC/AT PORTABLE **$1999**
Options:
Tech PC/XT Portable with 20 MB Hard Disk $2399
Tech PC/XT Portable with 1200 Baud Internal Hayes Compatible Modem and 20 MB Hard Disk $2599

All TECH PC PORTABLES available with tape backups, hard disks up to 1 gigabyte, networking systems, and hundreds of other hardware and software accessories.

TECH PC

714/754-1170

2131 South Hathaway, Santa Ana, California 92705
Telex 272006 Answer Back - TECH FAX: 71415568325

CIRCLE NO. 235 ON READER SERVICE CARD FOR DESKTOP 286

Finally, a language worth

For years BASIC has been everyone's first language. And for almost as long, they've been tempted by other languages. Lured by promises of more speed, more power.

We have a solution. A new language that's a substantial improvement over BASICA. Faster. More structured. Finally, a compelling reason to leave BASIC.

Introducing Microsoft's QuickBASIC Compiler, Version 2.0.

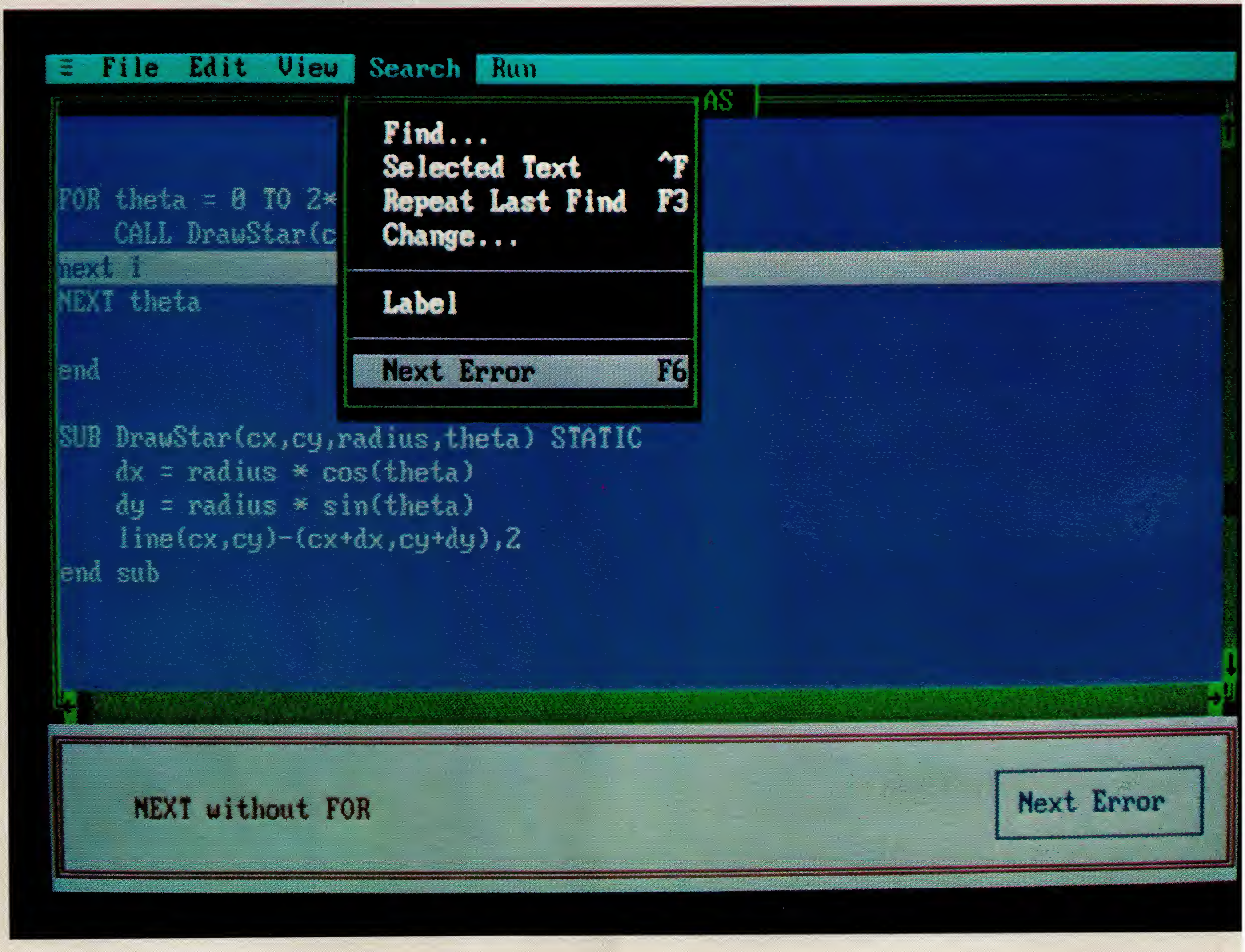
At last, you can have the latest programming techniques, combined with the solid foundation of BASIC. Our new compiler is as compatible with BASICA as you can get. At the same time it offers the extra speed and power you've been looking for.

Run faster with compiled code.

If there's one thing you've asked for, it's speed. And Microsoft® QuickBASIC simply blazes. Old BASICA programs will run up to ten times quicker once they've been compiled. Sometimes even faster.

Everything you need. Built-in.

Making programs run faster is only part of the story, though. The new Microsoft QuickBASIC Compiler includes a full-screen editor, built-in. So now you can make the jump from writing to RUNning in no time flat. Edit your program, compile it, and run it. Faster than any other BASIC compiler around. All without leaving our on-line help and prompts.



leaving BASIC for.

On the rare chance your program doesn't run 100% the first time out, we've got another surprise for you. The Microsoft QuickBASIC debugger. Our full-screen tracing lets you debug your programs while watching the source code execute. A line at a time, or with breakpoints. As easy as can be.

Our compiler is also smart enough to save you time. First, by finding any errors in one pass. Second, by putting your editor's cursor on the problem. Automatically. So you don't have to get lost in a maze of error codes and line-numbers.

The BASIC virtues. And more.

Speaking of line numbers, let's not. Because line numbers are strictly optional. And Microsoft QuickBASIC lets you use alphanumeric labels as well. Now you can GOTO *ErrorCheck* instead of line number 6815.

Or you could stop using GOTOs altogether. There are a variety of options that could make the GOTO an endangered species. Features like multi-line IF-THEN blocks. And named subprograms. Now your BASIC programs can be as structured and organized as you want.

We've only just begun to talk about the virtues of Microsoft QuickBASIC. There are dozens of enhancements to your favorite language. Things like larger arrays. Local and global variables. Reusable modules that let you create libraries of your most often-used routines. All explained in a revised manual that includes a complete language reference.

Making your quick escape.

If all these features follow your BASIC instincts, then zip on down to your nearest Microsoft dealer. That's where you'll discover the best surprise of all. The price. Only $99 for the best reason to leave BASIC.

For the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada, call (416) 673-7638.

Microsoft® QuickBASIC
The High Performance Software™

Microsoft QuickBASIC Compiler Version 2.0 for IBM® PC and Compatible Computers.

BASICA Compatibility
- Sound statements including SOUND and PLAY.
- Graphics statements including WINDOW, VIEW, DRAW, GET, PUT, LINE, CIRCLE, LOCATE and SCREEN.
- Support of EGA extended graphics modes. NEW!
- BASICA structures are supported including WHILE/WEND, IF/THEN/ELSE, FOR/NEXT, GOSUB/RETURN, and event handling.

Results of Sieve Benchmark	BASICA 3.1	Microsoft QuickBASIC 2.0
Seconds per iteration	78	0.52

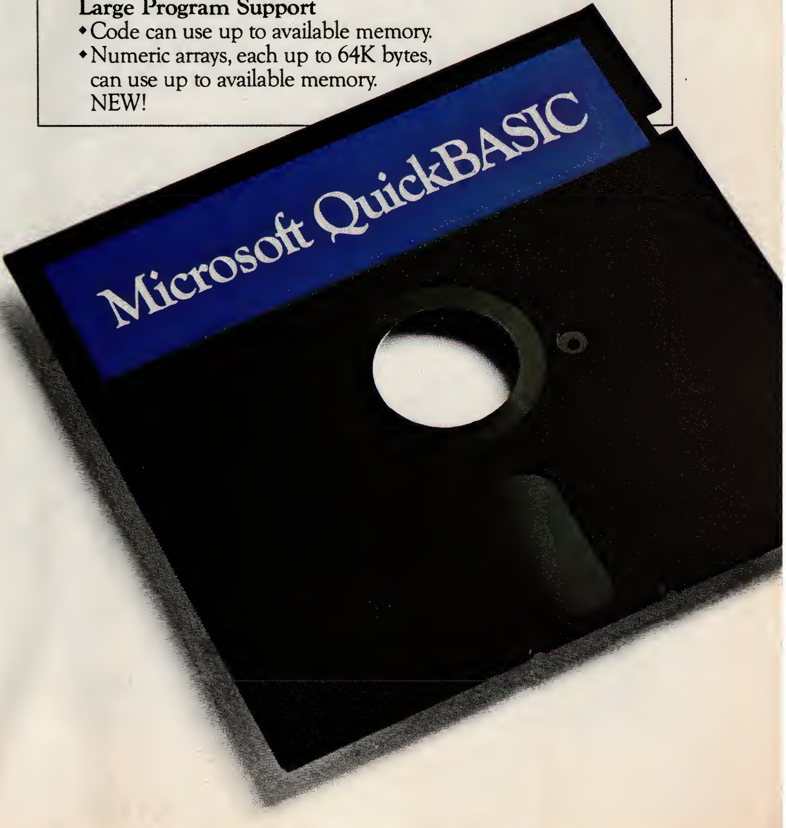
Complete Programming Environment
- Built-in Editor that places the cursor on found errors automatically. NEW!
- Compile entirely in memory at speeds up to 6000 lines per minute. NEW!
- Link routines once when starting a programming session and no need to link again when changing programs. NEW!
- Built-in debugger with single-step, animate, and trace modes. NEW!
- Create stand-alone programs.

Alphanumeric Labels
- Can be used to make your programs more readable. Line numbers are not required but are supported for BASICA compatibility.

Structured Programming Support
- Block IF/THEN/ELSE/END IF eliminates the need for GOTO statements. NEW!
- Subprograms can be called by name and passed parameters. Both local and global variables are supported.

Modular Programming Support
- Separate compilation allows you to create compiled BASIC libraries to use and re-use your programs.
- A library of routines to access DOS and BIOS interrupts is supplied. NEW!

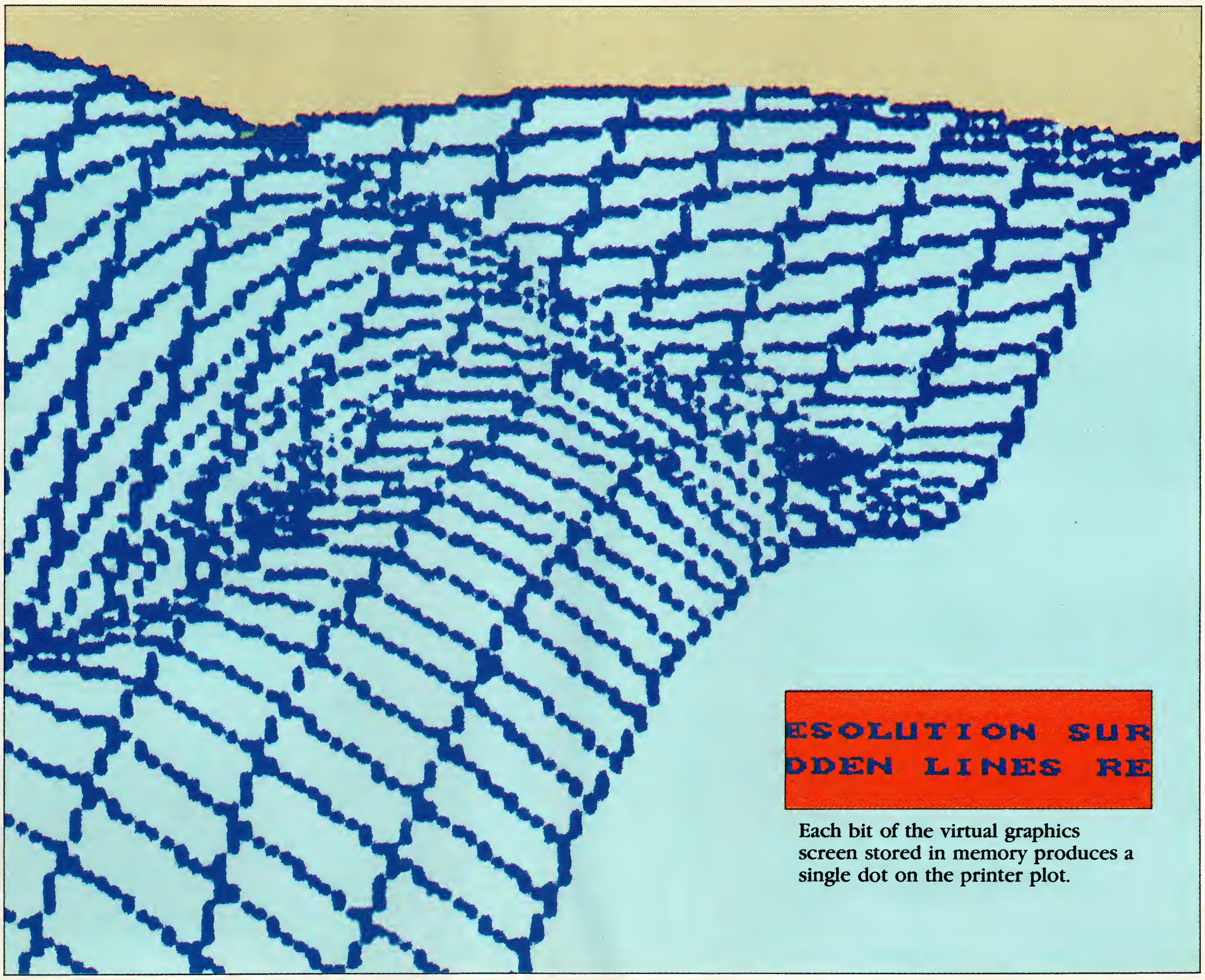
Large Program Support
- Code can use up to available memory.
- Numeric arrays, each up to 64K bytes, can use up to available memory. NEW!



A Virtual Graphics Screen

High-resolution graphics output to a printer is possible using a virtual screen in memory. This program lays the groundwork for individual user designs.

RICHARD CHANDLER and GARY FAULKNER



Each bit of the virtual graphics screen stored in memory produces a single dot on the printer plot.

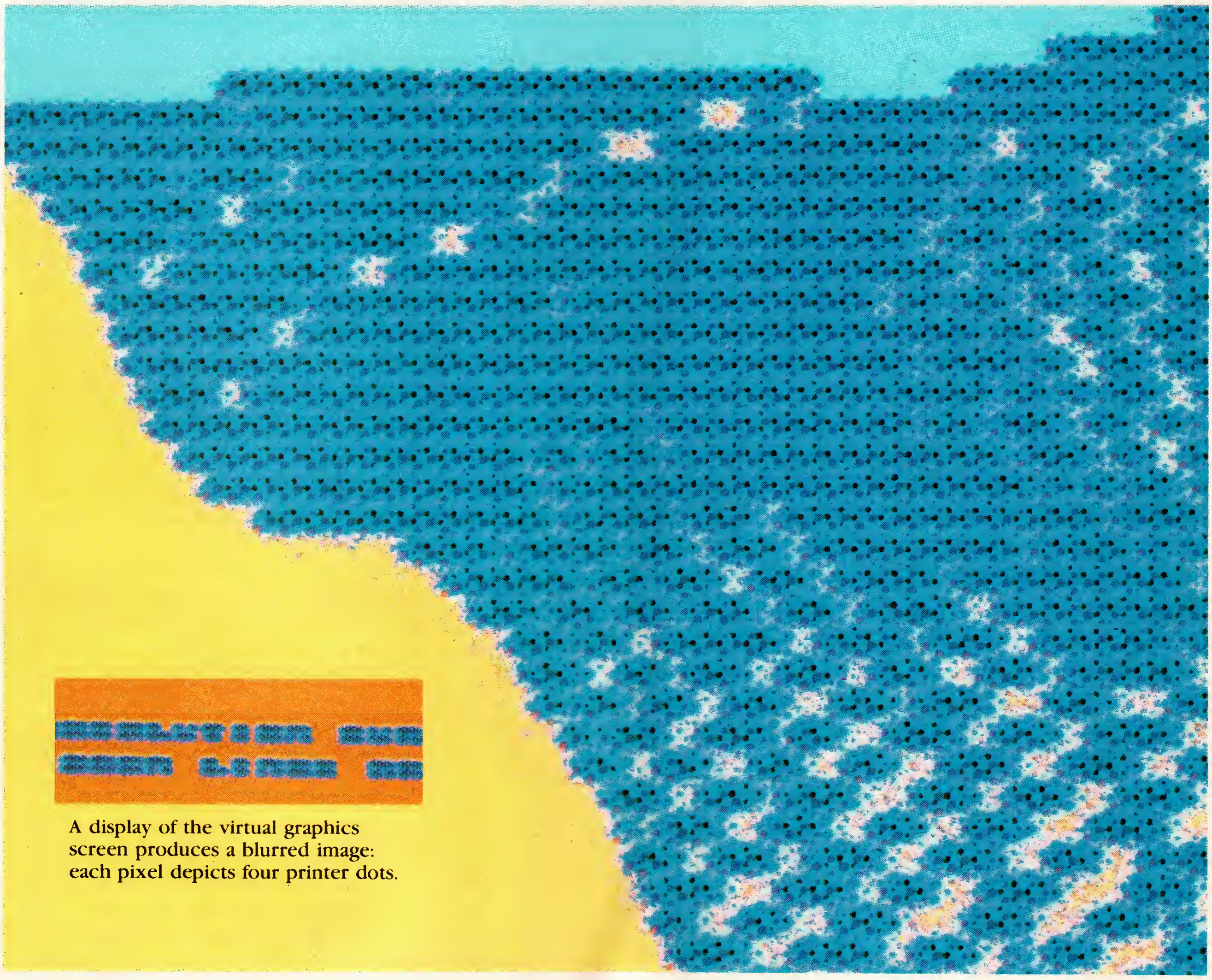
The IBM PC graphics screen is a little smaller than a standard 8½-by-11-inch sheet of paper, yet a dump of it to the printer always looks coarse. This is because one dot on the printer is much smaller than one pixel on the (high-resolution) screen. To make the two images end up approximately the same size, GRAPHICS.COM (the screen dump program included with DOS) causes each screen dot to be replicated four times vertically by the printer. It would be preferable, however, to put the finer resolution of the printer to better advantage.

One solution is to use a portion of memory as a virtual screen with a resolution similar to that of the printer. VSCREEN.PAS (listing 1) illustrates the principle involved. The Auto_Draw section is designed for customization by the user to provide the required draw capabilities. Note that this program occupies about 70KB of RAM as it is—a custom draw program will add to this amount. This obviously reduces the amount of RAM available to store the virtual graphics screen. For example, a complex surface plot with hidden line removal using this technique might require 160KB. Two examples of the printer plots possible are shown in figure 1. (The changes necessary to incorporate the SURFACE.PAS hidden-line program—as described in "The Painter's Algorithm," Richard Chandler and Gary Faulkner, November 1985, p. 181—into the Auto_Draw section of VSCREEN.PAS are shown in table 1. To use the full resolution of the EGA, further modification is necessary.)

When in optimum graphics mode for this application, the IBM Graphics Printer and Proprinter are capable of 960 dots across the page and 72 dots per inch vertically. The procedures here (implemented in Turbo Pascal) provide for a 640-by-800-dot virtual screen, for a total of 512,000 dots. This is four times the 128,000 dots of the high-resolution graphics screen. Each byte of memory holds eight dots, so the program has a virtual screen requirement of 64,000 bytes. Because the largest array addressable by Turbo Pascal can contain no more than MaxInt (32,767) cells, this virtual screen consists of two (two-dimensional) arrays of 32,000 bytes—**screen[0]** and **screen[1]**. To avoid the 64KB limitation on the data segment, they are stored in the heap.

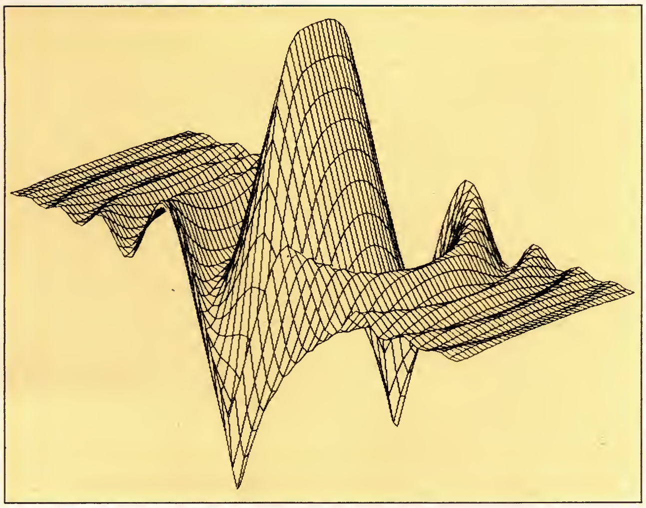
The procedures and functions described below provide the tools for a basic implementation. The program is driven by a menu on the monitor screen that lists seven procedures:

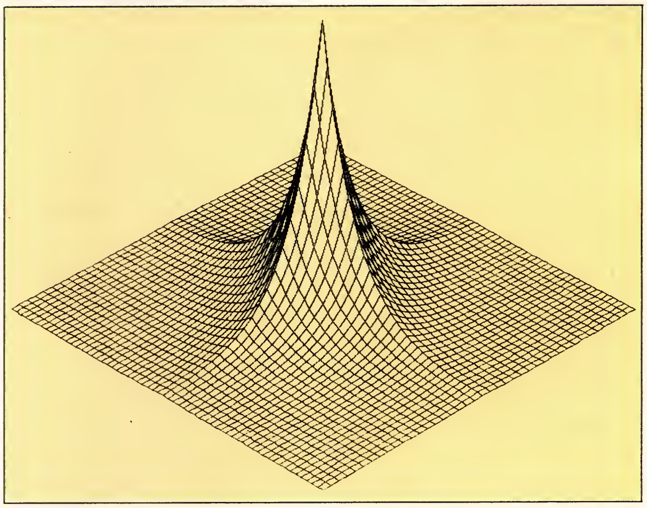
Auto_Draw. A user-supplied procedure for programmed drawing, Auto_Draw allows the user to develop a library of Auto_Draw modules and import a particular one as desired. For example, one module could graph functions of



A display of the virtual graphics screen produces a blurred image: each pixel depicts four printer dots.

FIGURE 1: *Typical Printer Plots*





These two printer plots (reduced here) were produced using VSCREEN.PAS with SURFACE.PAS integrated into Auto_Draw.

the form $y = f(x)$, another could make surface plots. A module such as the latter is described below (it is a simple adaptation of SURFACE.PAS).

In the VSCREEN.PAS listing, Auto_Draw is null. To plot an image properly, the actual coordinates must be scaled to the virtual screen coordinates. If $xmin \leq x \leq xmax$ and $ymin \leq y \leq ymax$ are actual coordinates, the proper virtual screen coordinates a and b are as follows:

```
a = round(639*(x - xmin) / (xmax -
  xmin))
b = round(799*(y - ymin) / (ymax -
  ymin))
```

Write_to_Screen. The user can place text on the monitor screen using Write_to_Screen. It shows 200 lines (of the 800 available) and a cursor that indicates where the text will appear. If the cursor is moved to the top or bottom of the monitor screen, it causes the virtual screen to scroll past. The monitor screen can be thought of as a window into the virtual screen, showing one-fourth of it at a time.

The character set used to write to the virtual screen is the same one used by the BIOS to write to the actual graphics screen. (This is an array of 1,024 bytes beginning at address segment F000H offset FA6EH). When the virtual screen is printed, these characters appear small, but they are well formed and perfectly legible. This may seem a questionable programming practice because no guarantee says that addresses in future versions of the BIOS will remain the same. It is, however, a harmless exercise; the worst that could happen is that it would put garbage on the screen. The address could be changed easily to a new value. This procedure also can be used to erase portions of the screen by inserting blank text (the space character).

Draw_Line. Draw_Line places lines and points on the virtual screen. A cursor that locates the initial and terminal points of the line can be moved in large steps or small steps, as controlled by toggling the – (minus) key on the right side of the IBM keyboard. The virtual screen can be viewed as a whole (by collapsing four lines to one) or in 200-line, high-resolution pieces; this is toggled using the + (plus) key.

When the cursor is at one end of the desired line, press the Enter key, move the cursor to the other end of the line, and press Enter again. This draws the line on the virtual screen. To place a point, press Enter twice without moving the cursor. High-resolution mode is suggested for placing the end points precisely; cursor position is only approximate in low resolution.

After the initial point of a line has been entered, the line is drawn and undrawn on the actual screen as the cursor is moved. If it moves over an existing feature on the screen, it may erase portions of that feature, but this happens only on the monitor screen, not on the virtual screen. The correct image can be recovered by toggling between high and low resolution.

View_Screen. The View_Screen procedure shows the whole virtual screen collapsed vertically four lines to one, which is accomplished by ORing the four points together.

Print_Screen. The virtual screen is dumped to the IBM Graphics Printer or the Proprinter (or any printer graphics compatible to these) with Print_Screen.

Disk. The Disk procedure saves or retrieves a virtual screen from the disk.

Clear_Screen. Clear_Screen clears the virtual screen by filling it with zeros.

The above menu of drawing procedures is achieved through the following subprograms within VSCREEN.PAS:

Inkey. The deficiencies in Turbo Pascal's Keypressed function are overcome in Inkey. Keypressed does not return the pressed key nor does it reset to be used again. Hi(Inkey) returns the keyboard scan code and Lo(Inkey) returns the ASCII code of the struck key (refer to the IBM *Technical Reference* for further explanation). Inkey removes this information from the keyboard buffer so it can be used repeatedly.

Point(a,b,c). This function places a pixel of color c at integer coordinates (a,b), where $0 \leq a \leq 639$, $0 \leq b \leq 799$. When c is equal to 0, the pixel is the background color (the white paper); if c is not equal to 0, the color is the foreground color (the black printer ink). If a or b is out of range, Point (a,b,c) does nothing. Although it could have been written using ordinary Pascal statements, this procedure was created using an Inline procedure to make it as fast as possible. It is the workhorse upon which everything else depends.

Point. Point first creates a mask with 1 in the correct bit position of the byte and 0 elsewhere. It then computes the address of the appropriate byte in virtual screen memory that contains the point (a,b), and, if c is equal to 1, ORs the memory byte with the mask byte. If c is equal to 0, the mask bits are inverted, thus a 0 is in the correct bit

TABLE 1: *Changes to SURFACE.PAS*

PROGRAM/PROCEDURE	CHANGE/DELETE
SURFACE.PAS	Change **Program SURFACE;** to **Procedure Auto_Draw;**
Procedure SCALE_TO_SCREEN	Change **dz := (zmax - zmin)/199;** to **dz := (zmax - zmin)/799;**
	Change **q[i,j] := 199 - round ((z[i,j] - zmin)/dz);** to **q[i,j] := round ((z[i,j] - zmin)/dz);**
Procedure BLANK(y)	Change **line(x3,x4,y0);** to **line (x3,y0,x4,y0,0);**
Procedure DRAWBOX	Change the four **draw** statements to **line** statements
Procedure LINE(x0,x1,y);	Delete procedure
Procedure GRAPH	Delete **HiRes; HiResColor (10);**
MAIN PROGRAM	Delete **repeat until keypressed;**
	Delete **TextMode(3);**
	Change **end.** to **end;**

These modifications allow the program SURFACE.PAS to be incorporated into VSCREEN.PAS as an example of a typical application for the Auto_Draw procedure.

position and the remainder of the bits show 1. The memory byte is ANDed with the mask byte.

Line(a,b,p,q,c). Bresenham's algorithm draws the line in color *c* from (*a,b*) to (*p,q*). Clipping occurs for parts of a line not on the screen. This algorithm is very fast; it uses only integer arithmetic.

View_Partial_Screen(i). This function puts 200 rows of the virtual screen, beginning at row i ($0 \leq i \leq 599$), onto the monitor. For speed, an Inline process is used (as with View_Whole_Screen).

Extensions and modifications to these procedures are possible. For a printer not graphics compatible with the IBM printers, Print_Screen *must* be modified. Moreover, the choice of printer may indicate more extensive changes. The exceptional resolution of the Toshiba 1340 would permit a 1,600-by-1,280 screen to be printed on a single sheet of paper. The Okidata 92 uses only seven bits of graphics information per print operation. The program could be modified to graph only on seven bits of a byte, which would waste an eighth of the virtual screen memory. Alternatively, Print_Screen could be changed extensively to map seven bits at a time and no additional virtual screen memory would be necessary.

As indicated above, the character set in BIOS is used by Write_to_Screen to produce small characters. If larger characters are required, a separate set could be defined and Write_to_Screen changed accordingly. A more flexible erase procedure (other than typing blank characters) also could be implemented. A reference grid would be helpful when precise placement of points and/or lines is necessary; this could be arranged so as to appear only on the monitor screen. Many other extensions are possible—the code was kept simple to provide an uncomplicated base for customization.

Richard Chandler and Gary Faulkner both hold doctorates in mathematics. They currently are teaching mathematics at North Carolina State University in Raleigh.

LISTING 1: *VSCREEN.PAS*

```
Program VSCREEN;

{--------DECLARATIONS-----------------------------------------}

type
   scrptr   =    ^scr;
   scr      =    array[0..399,0..79] of byte;
   letter   =    array[0..7] of byte;
   row      =    string[80];
 var
   j        :    integer;
   screen   :    array[0..1] of scrptr;
   char_set :    array[0..127] of letter absolute $F000:$FA6E;


{--------EXTERNAL PROCEDURES FROM TURBO 3.0---------------}

procedure Graphics;                                   external 'GRAPH.BIN';
procedure HiRes;                                      external Graphics[6];
procedure HiResColor(Color: Integer);                 external Graphics[9];
procedure Plot(X,Y,Color: Integer);                   external Graphics[21];
procedure Draw(X1,Y1,X2,Y2,Color: Integer);           external Graphics[24];
procedure GetPic(var Buffer;X1,Y1,X2,Y2:Integer);external Graphics[36];
procedure PutPic(var Buffer; X,Y: Integer);           external Graphics[39];


{--------RETURNS SCAN/ASCII CODE FOR KEY PRESSED----------}

Function Inkey : integer;
begin
  inline
      ($33/$C0/          { XOR AX,AX       0 to AX            }
       $CD/$16/          { INT 16H         Read Kbd into AX   }
       $89/$46/$04);     { MOV [BP+04],AX  Return sc/asc code }
end;


{--------CLEARS VIRTUAL SCREEN--------------------------------}

Procedure Clear_Screen;
begin
   FillChar(screen[0]^,32000,0);
   FillChar(screen[1]^,32000,0);
end;


{--------PLOTS POINT (x,y) ON VIRTUAL SCREEN in COLOR c----}

Procedure Point(x,y,c:integer);
var
   a,b : integer;
begin
   if ( x >= 0 ) and ( x <= 639 ) and ( y >= 0 ) and ( y <= 799 ) then
   begin
   a := Seg(screen[0]^);
   b := Ofs(screen[0]^);
   Inline($8B/$4E/$08/    { MOV  CX,[BP+8]    x -> CX                }
          $51/            { PUSH CX           Save x on Stack        }
          $80/$E1/$07/    { AND  CL,07        x MOD 8 -> CL          }
          $BB/$80/$00/    { MOV  BX,0080      128 -> BX              }
          $D3/$EB/        { SHR  BX,CL        Shift it right         }
          $8B/$46/$06/    { MOV  AX,[BP+6]    y -> AX                }
          $BA/$50/$00/    { MOV  DX,50H       80 -> DX               }
          $F7/$E2/        { MUL  DX           80*y -> AX             }
          $5A/            { POP  DX           x -> DX                }
          $B1/$03/        { MOV  CL,03        3 -> cl                }
          $D3/$EA/        { SHR  DX,CL        x div 8 -> DX          }
          $01/$D0/        { ADD  AX,DX        80*y + x div 8 -> AX   }
          $03/$46/$FA/    { ADD  AX,[BP-6]    Byte addr -> AX        }
          $93/            { XCHG AX,BX        Mask -> AX,Addr -> BX  }
          $1E/            { PUSH DS           Save DS                }
          $8B/$4E/$FC/    { MOV  CX,[BP-4]    Seg -> CX              }
          $8E/$D9/        { MOV  DS,CX        Seg -> DS              }
          $8B/$4E/$04/    { MOV  CX,[BP+4]    color -> CX            }
          $E3/$04/        { JCXZ              Jump +4 if color 0     }
          $09/$07/        { OR   [BX],AX      Set Point              }
          $EB/$04/        { JMP  +4           Jump to end            }
          $F7/$D0/        { NOT  AX           Flip mask              }
          $21/$07/        { AND  [BX],AX      Blank point            }
```

```
        $1F);          { POP  DS         Recover DS            }
   end;
end;


{--------DRAWS LINE (x1,y1) to (x2,y2) in COLOR c----------}

Procedure LINE(x1,y1,x2,y2,c : integer);
var
   i,j,k,dx,dy,ix,iy  :  integer;
begin
   if x1 < x2 then j := 1 else j := -1;
   if y1 < y2 then k := 1 else k := -1;
   dx := abs(x1-x2); dy := abs(y1-y2);
   if dx >= dy then
   begin
      iy := dx div 2;
      for i := 0 to dx do
      begin
         point(x1,y1,c);
         iy := iy+dy;
         if iy > dx then
         begin
            y1 := y1+k;
            iy := iy - dx;
         end;
         x1 := x1 + j;
      end;
   end
   else
   begin
      ix := dy div 2;
      for i := 0 to dy do
      begin
         point(x1,y1,c);
         ix := ix+dx;
         if ix > dy then
         begin
            x1 := x1+j;
            ix := ix - dy;
```

```
            end;
            y1 := y1 + k;
         end;
      end;
end;


{--------WRITES TO VIRTUAL SCREEN--------------------------}

Procedure Screen_Write(phrase:row; x,y:integer);
var
   i,j,k : integer;
begin
   i := y div 400;
   y := y mod 400;
   x := x div 8;
   for j := 1 to length(phrase) do
      for k := 0 to 7 do
         screen[i]^[y+k,x+j-1] := char_set[ord(copy(phrase,j,1)),7-k];
end;


{--------VIEWS WHOLE SCREEN--------------------------------}

Procedure View_Whole_Screen;
var
   a,b : integer;
begin
   a := Seg(screen[0]^);
   b := Ofs(screen[0]^);
   Inline(
      $B8/$00/$B8/       { MOV AX,B800H    Even Segment to   }
      $8E/$C0/           { MOV ES,AX          ES             }
      $BA/$00/$BA/       { MOV DX,BA00H    Odd Segment to DX }
      $BF/$F0/$1E/       { MOV DI,1EF0H    End Segment to DI }
      $8B/$76/$FA/       { MOV SI,[BP-6]   VScr offset to SI }
      $8B/$46/$FC/       { MOV AX,[BP-4]   VScr segment to   }
      $1E/               { PUSH DS           (Save DS)       }
      $8E/$D8/           { MOV AX,DS            DS thru AX   }
      $55/               { PUSH BP         Save BP           }
```

```
        $06/               { LP1: PUSH ES    Exchange          }
        $8E/$C2/           { MOV ES,DX         odd/even        }
        $5A/               { POP DX              Segments      }
        $BB/$28/$00/       { MOV BX,28H      words/row to BX   }
        $31/$ED/           { LP2: XOR BP,BP   0 to BP          }
        $31/$C0/           { XOR AX,AX        0 to AX          }
        $B9/$04/$00/       { MOV CX,04        4 to CX          }
        $3E/$0B/$02/       { LP3: OR AX,DS:[BP+SI] Combine 4 L }
        $83/$C5/$50/       { ADD BP,50H      Inc BP by 80      }
        $E2/$F8/           { LOOP LP3          4 times         }
        $26/$89/$05/       { MOV ES:[DI],AX  Move to Screen    }
        $46/$46/           { INC SI INC SI   Inc VScr offset 2 }
        $47/$47/           { INC DI INC DI   Inc Scr offset 2  }
        $4B/               { DEC BX          Dec word counter  }
        $75/$E7/           { JNZ LP2         Loop if not 0     }
        $81/$C6/$F0/$00/   { ADD SI,F0H      Inc VScr offset   }
        $81/$FA/$00/$BA/   { CMP DX,BA00H    Odd ScrSeg in DX? }
        $75/$03/           { JNE LP4         No:  Jmp to LP4   }
        $83/$EF/$50/       { SUB DI,50H      Yes: Dec DI by 80 }
        $83/$EF/$50/       { LP4: SUB DI,50H Dec DI by 80      }
        $73/$CE/           { JNC LP1          Jmp to LP1:DI < 0 }
        $5D/               { POP BP          Retrieve BP       }
        $1F);              { POP DS          Retrieve DS       }
end;


{--------VIEWS VIRTUAL SCREEN, LINE i to 200+i-------------}

Procedure View_Partial_Screen(i:integer);
var
   a,b : integer;
begin
   a := Seg(screen[0]^);
   b := Ofs(screen[0]^);
   Inline(
        $8B/$76/$FA/       { MOV SI,[BP-6]      b to SI        }
        $8B/$46/$04/       { MOV AX,[BP+4]      i to AX        }
        $B9/$50/$00/       { MOV CX,50H         80 to CX       }
        $F7/$E1/           { MUL CX             # bytes + b    }
        $01/$C6/           { ADD SI,AX          = Start Offset }
        $B8/$00/$B8/       { MOV AX,B800      Segment of odd }
        $8E/$C0/           { MOV ES,AX        to ES          }
        $BA/$00/$BA/       { MOV DX,BA00      Even Seg to DX }
        $8B/$46/$FC/       { MOV AX,[BP-4]    a TO AX        }
        $1E/               { PUSH DS          Save DS        }
        $8E/$D8/           { MOV DS,AX        a to DS        }
        $B8/$40/$1F/       { MOV AX,1F40H     Offset to AX   }
        $50/               { PUSH AX          Save Offset    }
        $BB/$64/$00/       { MOV BX,64H       100 to BX      }
        $06/               { HERE: PUSH ES    Exchange       }
        $8E/$C2/           { MOV ES,DX          odd/even     }
        $5A/               { POP DX               Segments   }
        $B9/$50/$00/       { MOV CX,50H       80 to CX       }
        $58/               { POP AX           Recover Offset }
        $29/$C8/           { SUB AX,CX        Dec Offset     }
        $50/               { PUSH AX          Save Offset    }
        $8B/$F8/           { MOV DI,AX        Offset to DI   }
        $F3/$A4/           { REP MOVSB        Move Row       }
        $5F/               { POP DI           Recover Offset }
        $57/               { PUSH DI          Save it        }
        $06/               { PUSH ES          Exchange       }
        $8E/$C2/           { MOV ES,DX          odd/even     }
        $5A/               { POP DX               Segments   }
        $B9/$50/$00/       { MOV CX,50H       80 to CX       }
        $F3/$A4/           { REP MOVSB        Move Row       }
        $4B/               { DEC BX           Dec row counter}
        $75/$E3/           { JNZ HERE         Loop           }
        $58/               { POP AX           Restore Stack  }
        $1F );             { POP DS           Restore DS     }
end;


{--------MOVES GRAPHICS CURSOR----------------------------}

Procedure Move_Cursor(var x,y,j,s:integer);
var
   cursor_buffer : array[0..24] of byte;
begin
   GetPic(cursor_buffer,x,y,x+8,y-8);
   draw(x,y-4,x+1,y-4,1); draw(x+7,y-4,x+8,y-4,1);
```

```
   draw(x+4,y,x+4,y-1,1); draw(x+4,y-7,x+4,y-8,1);
   j := Hi(Inkey);
   PutPic(cursor_buffer,x,y);
   case j of
      71 : begin
              y := y-s; x := x-s;
           end;
      72 : y := y-s;
      73 : begin
              y := y-s; x := x+s;
           end;
      74 : if s = 1 then s := 8 else s := 1;
      75 : x := x-s;
      77 : x := x+s;
      79 : begin
              y := y+s; x := x-s;
           end;
      80 : y := y+s;
      81 : begin
              y := y+s; x := x+s;
           end;
   end;
   if x < 4 then x := 4 else if x > 635 then x := 635;
   if y < 4 then y := 4 else if y > 203 then y := 203;
end;
{--------DRAWS LINE--------------------------------------}

Procedure Draw_Line;
var
   x,y,xs,ys,lx,ly,vy,i,j,s,k  :  integer;
   start,stop,flag             :  Boolean;
begin
   textcolor(white);
   gotoxy(10,5); write(' F6 : Begin        Esc : Return to Main Menu ');
   textcolor(yellow);
   gotoxy(14,12); write('Cursor Keys are Active');
   gotoxy(14,14); write('  - : Toggles Cursor between Fast and Slow');
   gotoxy(14,16); write('  + : Toggles between Hi and Lo Resolution');
   gotoxy(14,18); write('<á> : Begins or Ends Line');
   repeat
```

```
      k := Hi(Inkey);
   until (k  = 64) or (k = 1);
   if k = 1 then exit;
   s := 8; x := 320; y := 100; i := 300;
   start := false; stop := false; flag := true;
   HiRes; HiResColor(LightGreen);
   repeat
     if flag then
     begin
        View_Whole_Screen;
        if start then ys := 199 - vy div 4;
        y := 199 - (199-y+i) div 4;
        if y < 0 then y := 0 else if y > 199 then y := 199;
     end
     else
     begin
        i := 4*(199-y) - 100;
        if i < 0 then i := 0;
        if i > 600 then i := 600;
        View_Partial_Screen(i);
        if start then ys := 199-vy+i;
        y := 199 - 4*(199-y) + i;
        if y < 0 then y := 0 else if y > 199 then y := 199;
     end;
     repeat
        Move_Cursor(x,y,j,s);
        if j = 28 then
        begin
           if not start then
           begin
              xs := x+4; ys := y-4;
              lx := xs; ly := ys;
              if flag then vy := 4*(199-ys)
              else vy := i+199-ys;
           end;
           if start then stop := true;
           start := true;
        end;
        if start then
        begin
```

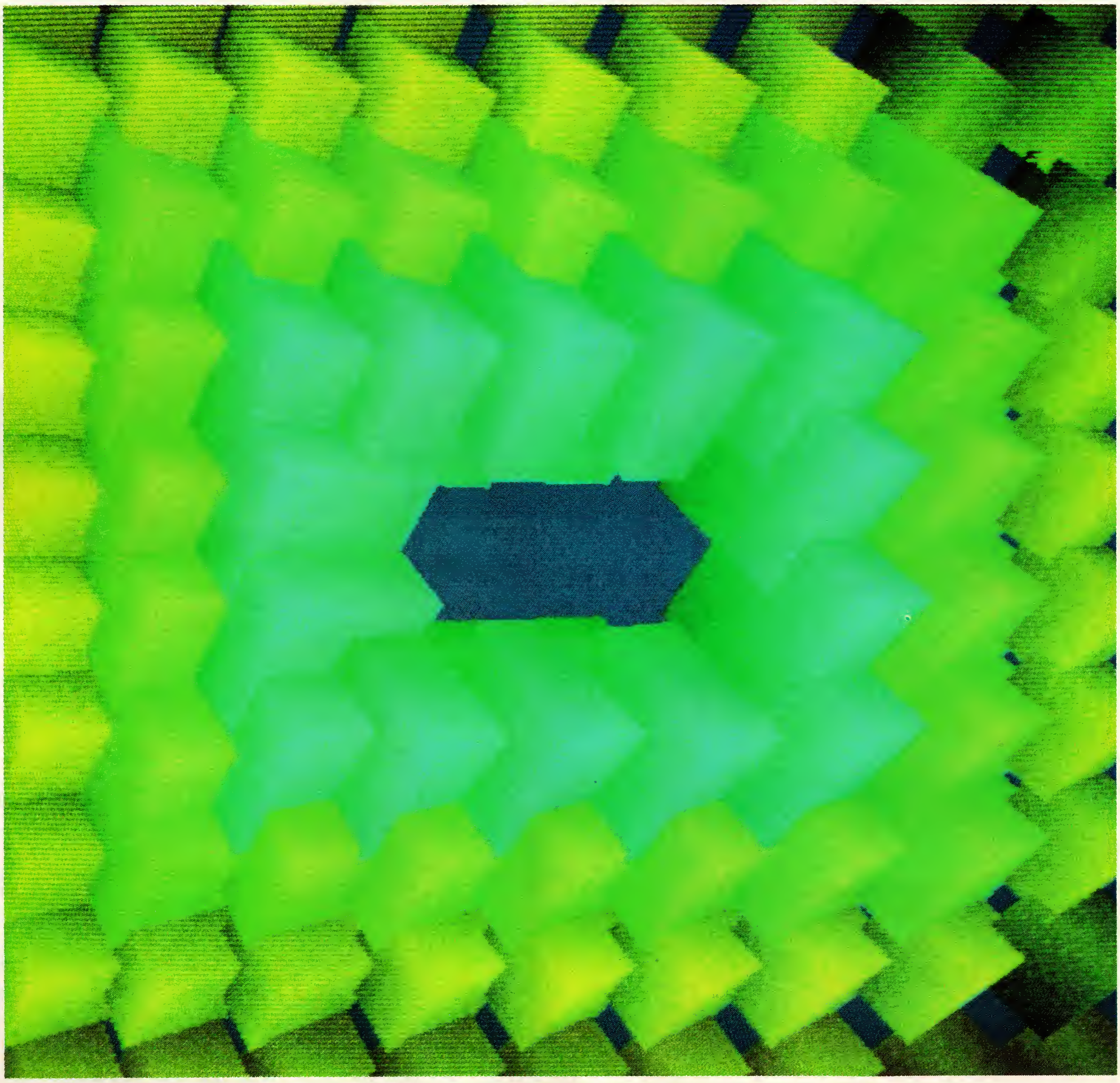
```
         draw(xs,ys,lx,ly,0);
         lx := x+4; ly := y-4;
         draw(xs,ys,lx,ly,1);

      end;
      if (j = 28) and (stop = true) then
      begin
         draw(xs,ys,lx,ly,1);
         if flag then ly := 4*(199-ly)
         else ly := i+199-ly;
         line(xs,vy,lx,ly,1);
         start := false; stop := false;
      end;
   until (j = 1) or (j = 78);
   if j = 78 then flag := not flag;
 until j = 1;
end;


{--------WRITES TO HIGH RESOLUTION SCREEN-----------------}

Procedure Write_to_Screen;
var
   x,y,i,j,k,l     : integer;
   ch              : char;
   flag            : Boolean;
   cursor_buffer   : array[0..24] of byte;
   phrase          : row;
begin
   textcolor(White);
   gotoxy(10,5); write('F4 : Begin       Esc : Return to Main Menu ');
   textcolor(Yellow);
   gotoxy(10,12); write('PgUp/PgDn Scrolls Up or Down ~ Screen');
   gotoxy(10,14); write('Use Cursor Keys to Position Phrase');
   gotoxy(10,16); write('Type  <âì To Begin and End Phrase' );
   repeat
      l :=  Hi(Inkey);
   until (l = 62) or (l = 1);
```


```
   if l = 1 then exit;
   x := 320; y := 103; i := 0; flag := true;
   HiRes; HiResColor(LightGreen);
   repeat
      if flag then View_Partial_Screen(i);
      GetPic(cursor_buffer,x,y,x+8,y-8);
      draw(x,y-8,x+8,y-8,1); draw(x+8,y-8,x+8,y,1);
      draw(x,y,x,y-8,1); draw(x,y,x+8,y,1);
      flag := false; j := Hi(Inkey);
      PutPic(cursor_buffer ,x,y);
      case j of
         28 : begin
                 gotoxy(1+(x div 8),1+((y-7) div 8));
                 read(phrase);
                 Screen_Write(phrase,x,i + (199-y));
              end;
         72 : y := y-8;
         75 : x := x-8;
         77 : x := x+8;
         80 : y := y+8;
         73 : i := i+50;
         81 : i := i-50;
      end;
      if (j = 73) or (j = 81) then flag := true;
      if x > 632  then x := 632
      else if x < 0 then x := 0;
      if y > 199 then
      begin
         y := 199;
         i := i-8;
         flag := true;
      end
      else if y < 7 then
      begin
         y := 7;
         i := i+8;
         flag := true;
      end;
      if i > 600 then i := 600
      else if i < 0 then i := 0;
   until j = 1;
end;


{--------SAVE OR RESTORE PICTURE--------------------------}

Procedure Disk;
var
   k        : integer;
   name     : string[14];
   iofile   : file;
begin
   ClrScr;
   gotoxy(10,10);write('F1   : Save Screen to Disk');
   gotoxy(10,12);write('F2   : Retrieve Screen From Disk');
   gotoxy(10,14);write('Esc  : Return to Main Menu');
   repeat
      k := Hi(Inkey)
   until k in [1,59,60];
   if k = 1 then exit;
   ClrScr; gotoxy(10,15);
   Write('File Name ? ');
   Readln(Name);
   assign(iofile,Name);
   case k of
      59 : begin
              rewrite(iofile);
              blockwrite(iofile,screen[0]^,500);
           end;
      60 : begin
              reset(iofile);
              blockread(iofile,screen[0]^,500);
           end;
   end;
   close(iofile);
end;


{--------PRINTS VIRTUAL SCREEN----------------------------}
```

```
Procedure Print_Screen;
const
   lf      =  #10;
   cr      =  #13;
   can     =  #24;
   esc     =  #27;
var
   b         :  byte;
   x, y, c  :  integer;
begin
   x := 0;
   write(Lst,can + esc + '3' + #24);      { Set Graphics line spacing }
   gotoxy(32,14); write('Printing  Row');
   gotoxy(32,16); write('Keypress to abort');
   repeat
      gotoxy(47,14); write('  ');
      gotoxy(47,14); write(80-x);
      write(Lst,'            ' + esc + 'L');{ Graphics mode on            }
      c := 800;
      repeat
         c := c-1;
      until (screen[c div 400]^[c mod 400,x] <> 0) or (c = 0);
      write(Lst,chr((c+1) mod 256),chr((c+1) div 256));
                                                  { # of graphic bytes }
      for y :=   to c do
      begin
         b := screen[y div 400]^[y mod 400,x];
         Inline($31/$D2/                 { XOR DX,DX    0 -> DX }
               $8B/$86/b/               { MOV AL,b     b -> AL }
               $30/$E4/                 { XOR AH,AH    0 -> AH }
               $CD/$17);                { INT 17H      Print b }
      end;
      write(Lst,lf + cr);               { Line feed/carriage return }
      x := x+1;
   until (x = 80) or (Keypressed);
   write(Lst,esc + '2');                { Restore text spacing      }
end;
{--------USER SUPPLIED AUTOMATIC DRAW ROUTINE----------------}

Procedure Auto_Draw;
begin
end;


{--------MAIN PROGRAM------------------------------------}

begin
   for j := 0 to 1 do new(screen[j]);
   Clear_Screen;
   repeat
      TextMode(3); ClrScr;
      gotoxy(15,5);textcolor(White);write('MAIN MENU');
      textcolor(Yellow);
      gotoxy(10,8);  write('F1   : Clear Screen');
      gotoxy(10,10);
         write('F2   : Auto Draw      (User supplied procedure)');
      gotoxy(10,12); write('F3   : Disk File Access');
      gotoxy(10,14); write('F4   : Write to Screen');
      gotoxy(10,16); write('F6   : Draw Line');
      gotoxy(10,18); write('F8   : View Screen');
      gotoxy(10,20); write('F10  : Print Screen');
      gotoxy(10,22); write('Esc  : Exit Program');
      j := Hi(Inkey);
      ClrScr;
      case j of
         59 : Clear_Screen;
         60 : Auto_Draw;
         61 : Disk;
         62 : Write_to_Screen;
         64 : Draw_Line;
         66 : begin
                 HiRes; HiResColor(LightGreen);
                 View_Whole_Screen;
                 repeat until Hi(Inkey) = 1;
              end;
         68 : Print_Screen;
      end;
   until j = 1;
end.
```

DATA MANAGERS AS DEVELOPMENT TOOLS

A Data Manager for End-user Development



COMPUTER GRAPHIC • DOV JACOBSON

Smooth, efficient, logical, and automatic, DataEase has been designed with the end-user interface firmly in mind.

DAVE BROWNING

A newcomer to the application development environment, the latest version of DataEase adds the dimension of application development to this menu- and form-driven, single-user data management system, earning it a place in *PC Tech Journal's* data manager series. Although it exhibits a few rough edges, the program's firm foundation in user interface and data entry/presentation provides easy implementation of sophisticated features that would require substantial programming in other data managers.

To its previously end-user-oriented program, Software Solutions brings multiview capability for data entry in related files, virtual fields, user-definable, context-sensitive help, 50 new functions, a data dictionary, structured procedural language, and a variety of other features and improvements. Because it was designed with the end-user interface in mind, DataEase has an abundance of functions that provide the application designer with tools to manage data validation and table look-ups efficiently and effectively. Attractive data entry screens, data presentation screens, and menus are easily created, and predefined reports can be associated with custom data entry forms for ad hoc selection of report query criteria.

DataEase is based on a mix of the relational and entity-relationship models of data representation; forms (files or tables) of data are linked through user-defined relationships between fields of equal value. Relationships may be either predefined or created on an ad hoc basis within report queries. Separate B-tree index files provide for rapid retrieval of data.

The program runs on IBM PCs, PC/XTs, PC/ATs, and compatibles under DOS and on Wang PCs and some other computers. DataEase uses between 384KB and 640KB of memory and the math coprocessor if it is available. A wide variety of printers is supported, and instructions are provided for tailoring the program to accommodate less common ones. While a dual-floppy-disk system can be used, a hard disk is highly recommended.

DataEase is delivered on four diskettes: system, utilities, demonstration, and tutorial. A diskette containing application templates is promised upon return of the registration card, but was not available when this article was being prepared. Software Solutions indicates in the manual that it soon will produce a *Catalog of Independent Software Vendors' Applications.*

The DataEase diskettes are not copy protected. Security protection is provided by user log-on with passwords assigned by the system administrator or developer. Up to seven security levels in three groups can be assigned to users in a file that can be encrypted for disk storage. User access to menus and fields is assigned by security group.

The manual provides instructions to make working copies of the system and utilities disks. DOS may be installed on the working floppy disk to create a bootable diskette.

Installation of DataEase on a hard disk is straightforward. All files from the system and utilities diskettes are copied into a subdirectory that then can be used for the DataEase programs as well as for databases; separate subdirectories should be created for database applications. Note that if the DataEase subdirectory is included in the DOS PATH, the user can invoke DataEase from other subdirectories.

A data subdirectory must be prepared for DataEase data by copying several files into it. This is accomplished by changing to the new data subdirectory and executing the utility program DECOPY before starting DataEase. This must be done for each data directory.

The DOS CONFIG.SYS file requires the FCBS=35,35 statement for DOS 3.0 and later. The DataEase stack size also can be increased at program execution time if stack overflows have occurred when trying to process forms or reports of many levels.

Once DataEase is started, the system configuration option from the main menu allows users to customize installation. The default program and system drives, display type, and printer installations are configured, users are defined, and screen styles selected. These values may be updated at any time.

Within a disk subdirectory, DataEase may define up to 26 databases (named *A* through *Z*). Each database is a collection of related files with the following limits applicable to each of the 26: 255 files per database, 65,535 records per file, 4,000 characters per record, 255 fields per record, 255 characters per field, 255 reports per database, 16 screens per form, and 255 B-tree indexes (one per field) per form. While DataEase's capacities may seem modest in comparison to the billions of records and thousands of fields boasted by other data managers, they should be sufficient for most applications.

No special facilities such as application generators, compilers, or runtime modules are available, although a developer's toolkit is expected to be released soon. Field definitions can be stored in a dictionary, making development easier, and batch-like command files can be used to install forms, reports, and data into user applications. DOS command, batch, and executable files are

accessible from a choice on the database maintenance menu, and external programs can be executed from custom application menus.

DISTINGUISHING FEATURE

DataEase is distinguished by its use of *forms*, which are the visual representation of data. Forms define database structures, establish data entry screens, produce reports, and store definitions of relationships, fields, menus, and a variety of other information. DataEase uses the term *form* to refer to both the screen representation as well as to the data file itself.

A file is defined by painting a form on the screen. This means that one form per file can be defined. An editor provides facilities to cut and paste or to copy areas of the screen, including fields as well as text labels or prompts. Special graphics characters enhance the form appearance, and single or double line borders (boxes) are drawn or erased by marking opposite corners with Alt-F10 keystrokes. To avoid disturbing existing text or fields, the insert mode should be off when a border is drawn or erased. Other special characters are integrated into screens by repeatedly selecting extended characters from a multipage menu. This becomes tedious for screens containing many graphics characters, but the process produces attractive screens.

A field is defined by pressing the F10-Field key when the cursor is positioned at the desired screen location for the start of the field. A default field name is derived from any text entered to the left of the field position. This can be overridden. Field names can be up to 20 characters in length (although the manual mistakenly states 25 characters may be used) and can contain words separated by blanks. A dictionary form can be established and field specifications entered that can be used on other forms within the database.

Following the selection of a name for the field, a series of questions is asked, and the acceptable answers are presented as menus horizontally across the top or vertically down the right side of the screen. These types of menus are common throughout DataEase.

Besides the usual Text, Number, Date, and Yes or No field types found in most data managers, DataEase also offers powerful field types with predefined formats for items found in many business databases: Numeric String, Time, Dollar, and Choice.

Text fields hold up to 255 characters of general data. Number fields may be specified as integer, fixed point, or floating point, with 14-digit accuracy. The decimal position may be specified in fixed or floating point, and DataEase automatically positions commas in fixed-point numbers. Date fields are formatted in MM/DD/YY format, but this may be changed to DD/MM/YY using the country configuration option in the Configuration form. The default 100-year date span is from 01/01/1901 to 12/31/2000, but this can be changed on the Configuration form by specifying a starting year in the range from 1 to 75. Date arithmetic is supported.

Numeric strings may be specified as unformatted, as a user-defined format, or as one of two DataEase predefined formats: either Social Security Number (NNN-NN-NNNN) or phone number ((NNN)-NNN-NNNN). Unformatted numeric strings can be 255 digits long, and formatted strings can contain as many as 40 characters, including the fixed punctuation characters. The dollar field is a special case of the fixed-point number type. Time fields are formatted as 24-hour HH:MM:SS, and time arithmetic is provided. Functions to manipulate and display date and time fields are discussed below.

The Choice field permits the specification of as many as 99 choices of up to 60 characters each. Upon data entry, these choices are presented on a menu. The user selects one either by number or by typing in the minimum number of characters necessary to identify the choice. The set of choices may be named and saved for use in other fields or forms in the database. The Choice field is efficient for both data entry and internal storage because only the two-digit choice is saved in the file. The Yes or No field is a special case of the Choice field with two predefined choices and is stored the same way.

The DataEase diskettes are not copy protected. Security protection is provided by user log-on with passwords assigned by the system administrator or developer.

After the field type is selected, the field length is requested, if appropriate. The developer may not abandon the field definition before answering all questions up to this point. This could be considered a minor inconvenience.

The remaining questions are optional and may be skipped to accept the default values or answered to specify powerful data validation criteria. These field options are listed below.

- The field can be specified for required entry. During data entry, DataEase will not permit a record to be entered (added or modified) until all required fields are filled (nonblank).
- The user can decide whether or not a field is to be indexed.
- The field can be specified as unique so that DataEase will prohibit the entry or modification of records that would cause duplication of the data in this field.
- A range check can be implemented for the data in the field. Two 190-character fields are provided to enter the lower- and upper-limit range check specifications. These can be as simple as a single constant or very elaborate and complex with nested functions and data look-up from related tables or other fields on the form. Range checking may be used on any field type including Choice and Yes or No fields and is not limited to numeric calculations. This powerful feature takes the place of extensive code required in other systems.
- The field can be specified as a derived field, which means that the data contents may be calculated, looked up from other data, sequenced by DataEase, or assigned a default value. Again, the derivation formula can be up to 190 characters in length, and can combine data from related tables. Automatic sequencing can be applied only to text or numeric string fields (but not to number fields). A default value, if specified, will be placed in the field for confirmation or modification at data entry time. The system date may be used as the default value by specifying the formula as ??/??/??.
- The developer can prevent the operator from overwriting a derived field. This option can be answered "Yes," "No," or "Yes, and do not save (virtual)." This third choice causes the data not to be stored in the file, which saves disk space at the expense of recalculation time. It also can be used to force the look-up to occur each time the record is read to reflect changes in the look-up table values.
- Seven security levels can be assigned to users; these levels can be used to specify which users have field view and write privileges.

DATAEASE OVERVIEW

DATAEASE, version 2.5
Software Solutions, Inc., 12 Cambridge Drive, Trumbull, CT 06611; 203/374-8000

Product description. DataEase is a menu- and form-driven, single-user data management system based on the entity-relationship model. The product uses a structured procedural language, yet it is optimized for the end user.
IBM PC environment. DataEase runs on any IBM PC, PC/XT, PC/AT, or 100-percent compatible running under DOS 2.0 or later with 384KB to 640KB of RAM and two floppy-disk drives or one floppy drive and a hard disk. Several near compatibles also are supported. DataEase can be used with a variety of printers and provides the capability for the user to define additional printer types.
Network support. Software Solutions, Inc. says that a network version currently is under development.
Copy protection. The product is not copy protected.
Documentation. A single-volume reference manual is provided with the product. It is divided into 10 chapters and 5 appendices and includes extensive examples. The documentation also includes a 32-page "Quick Reference Guide" and a pamphlet that discusses changes and upgrades for users working with previous versions of DataEase. A nine-lesson tutorial manual is provided with the tutorial disk.
User interface. DataEase uses a menu-driven user interface that provides forms for data entry, queries, and reports. A relational query language with structured procedural features is available for custom reports.

Help facilities. On-line, context-sensitive help is available using the function key Alt-F1. Most input is selected from menus of acceptable values.
File capacities. DataEase allows a maximum of 64KB records per file (form), 4,000 characters per record, 255 fields per record, 255 characters per field, 255 forms and reports per database, 26 databases per directory, 16 screens per form, and 255 B-tree indexes (one per field) per form.
Data entry. DataEase is designed with data-entry operations in mind. The user easily can develop attractive forms with elaborate and extensive data validation and look-up. File structures and field definitions are automatically created and modified through the forms. Field definitions can be stored in a dictionary.
Application development facilities. Batch-like command files can be used to install forms, reports, and data into the user's applications. DataEase does not provide application generators, compilers, or a runtime version. A separate toolkit that is designed to be used with application development is now under development.
Security. The user's name and password are required for log-on. Seven user security levels in three groups can be assigned by password to limit user access to fields and menus. A user file of names and passwords is encrypted for disk storage.
Access to system facilities. DOS commands and batch and executable files are accessible from the database maintenance menu. External program execution can be integrated into custom application menus.
Queries and reports. Simple file filtering is supported in form view; other queries are possible via Quick or Full reports. Predefined or ad hoc relationships are used in queries and reports. The DataEase Query Language uses structured procedural statements. Flexible output report formats are possible, including columnar with grouping, field per line, record-entry form, predefined templates, or free-form user defined. Multiple records are possible across a report (for example, multi-up labels).
Utilities. The product includes a dictionary to store field definitions. Data validation errors are logged to an exception file on data import. Menu selections allow the user to backup and restore the database.
Data compatibility. DataEase allows data to be input from DIF, dBASE II and III, Mail-merge, Variable Length (delimited), Fixed-Length, and Lotus 1-2-3 formats. Delimited fields can be selectively input by field name.
Distribution. Distribution of DataEase is through distributors and dealers.
Price. $600.
Support. Phone support is available.
—*Dave Browning*

- Help information can be defined for each field. DataEase automatically creates a window of appropriate size for the help screen. Alternatively, the developer can choose to specify the size and location of the window along with the help message. The user can set field help to be displayed automatically or only when requested.
- The last choice is the field display attribute. Four selections are available: normal fields are displayed in reverse video; highlight 1 displays in high intensity; highlight 2 displays in normal intensity to look like surrounding form text; and highlight 3 makes the field information invisible. These choices may be redefined on the Screen Styles form.

After the form has been defined, it may be canceled (aborted), saved, or printed. The form definition is saved as a table that includes the screen layout, field definitions, and form statistics such as record size and memory requirements. The table is reorganized to number fields from left to right and top to bottom of the screen, regardless of the order in which they were defined. This definition is saved in a DOS file with an extension of .DBA. The file name is made up of four characters from the form name followed by the database letter and a three-character sequence starting with AAA. The data file associated with a form has the same file name with an extension of .DBM. Index files have the same file name and extensions of .Inn, where *nn* is the number, in hexadecimal, of the field indexed. Figure 1 is a print-out of the Articles form used in the sample application. (The sample application was prepared by *PC Tech Journal* editors for this series on data managers. For a complete explanation see "Sample Application Specifications," August 1985, p. 48. That article and sample data files also are available on PCTECHline.)

Forms are easily modified with the same procedures used to create them.

FIGURE 1: *Articles Form Definition*

1 10 20 30 40 50 60 70 80

EDITORIAL INVENTORY—Article File

Title: ____

Author: __________________ ____________ Day Phone: ____________

Co-author: __________________ ____________

Booking: Volume _ Number __

Date: _________ ____ Commissioned: _

Article Type: Category: _________________

Department: _____________________

Date due: ________ | Date Received: ________

Sizes: Editorial: __ | Payments: Article: _____

Listings: __ | Bonus: __

Total: __ | Total: _____

1 10 20 30 40 50 60 70 80

FIELD DESCRIPTIONS

No.	Name	Type	Long	Reqd	Index	Unique	Derived	Rng Chk	Prevent	Record size	Record offset
1	Title	Text	60	Yes	Yes	No	No	No	No	60	5
2	Author Last Name	Text	18	Yes	Yes	No	No	No	No	18	65
3	Author First Name	Text	12	Yes	No	No	No	No	No	12	83
4	Day Phone	Num.String	12	No	No	No	Yes	No	Virt	10	142
	Field calculation formula : lookup Author "Home Phone"										
5	Co-author Last Name	Text	18	No	No	No	No	No	No	18	95
6	Co-author First Name	Text	12	No	No	No	No	No	No	12	113
7	Volume	Num.String	1	Yes	Yes	No	No	No	No	1	125
8	Number	Num.String	2	Yes	Yes	No	No	Yes	No	2	126
	Lower Range : 1										
	Upper Range : 12										
9	Month	Text	9	No	No	No	Yes	No	Virt	9	152
	Field calculation formula : spellmonth (number)										
10	Year	Num.String	4	No	No	No	Yes	No	Virt	4	161
	Field calculation formula : 1982+ Volume										
11	Commissioned	Choice	1	Yes	No	No	No	No	No	1	128
	Choice 1: T										
	Choice 2: F										
12	Category	Choice	17	Yes	Yes	No	No	No	No	1	129
	Choice 1: Product Review										
	Choice 2: Technical Article										
	Choice 3: Department										
	Choice 4: Tech Notebook										
13	Department	Choice	21	Yes	No	No	No	Yes	No	1	130
	Choice 1: Programming Practices										
	Choice 2: Directions										
	Choice 3: Legal Brief										
	Choice 4: Product of the Month										
	Lower Range : IF ("Department"=Category, Programming Practices, BLANK)										
	Upper Range : IF ("Department"=Category, Product of the Month, BLANK)										
14	Date due	Date	8	Yes	No	No	Yes	No	Virt	6	165
	Field calculation formula : lookup Issue Deadline										
15	Date Received	Date	8	No	Yes	No	No	No	No	6	131
16	Editorial Pages	Number	2	No	No	No	No	No	No	1	137
	Number Type : Integer										
17	Payment	Number	5	No	No	No	No	No	No	2	138
	Number Type : Integer										
18	Listing Pages	Number	2	No	No	No	No	No	No	1	140
	Number Type : Integer										
19	Bonus	Number	2	No	No	No	No	Yes	No	1	141
	Number Type : Integer										
	Lower Range : 5										
	Upper Range : 20										
20	Total	Number	2	No	No	No	Yes	No	Virt	1	171
	Number Type : Integer										
	Field calculation formula : Editorial Pages + Listing Pages										
21	Total1	Number	5	No	No	No	Yes	No	Virt	2	172
	Number Type : Integer										
	Field Calculation formula : Payment + Bonus										

Record size 142

Memory required for form: Text 1361, Fields 1313, Total 2674 bytes.

Attractive forms can be created and powerful data validation criteria can be defined for fields—all without programming.

The field definition can be recalled for modification or deletion by placing the cursor within the field and pressing F10-Field. Changes are made only to the form definition table in memory until the modified form is saved. Up to this point, the process may be aborted. When directed to save the modified form, DataEase asks if it is to be saved under a new name, in which case the original form remains unchanged. When the modified form is not given a new name by the user, the modifications apply to the original form. Copying data from the original form to the new form is optional.

The data records are reformatted if the modifications included adding or deleting fields or changing field types or field lengths. Deleted records are removed from the data and index files, blank fields are derived again, and existing records are validated. Sufficient space must be available on the disk to hold two copies of the data file during the reorganization. Errors are logged by record and field number to an exception file that may be read with a text editor. Index files are added, deleted, or updated if indicated by the changes, whether or not the data records are reorganized. This reorganization process may be initiated manually from the Forms Definition Menu.

RELATIONSHIPS

DataEase thrives on relationships. Its capabilities for defining and processing relationships are extremely powerful and flexible. Unlike other data managers, DataEase allows relationships to be defined between one master table and several secondary tables. Relationships also can be defined within a single file and nested (chained) to a depth of 30 levels. Data reporting from one-to-many relationships is supported, again an unusual feature in microcomputer data managers.

Relationships are used for look-up operations to support data entry and validation and for data retrieval in queries and reports. They may be predefined or created during reporting. Predefined relationships must be used for data entry operations, but the act of specifying the relationship criteria can be deferred until the data entry form is completed. If the criteria are not specified before the form is used, however, DataEase displays an "unknown relationship" error message.

Predefined relationships are stored as records in a special form, which is reached from a selection on the main menu. Each definition specifies the conditions for a linkage between two forms, normally data forms. Up to three pairs of fields with equal values can be selected to link the files.

Names can be assigned to predefined relationships to distinguish among multiple relationships between files. For example, the linkage between the sample application's Articles form and the Title/Author form to retrieve an article's author might be named "Article's Author" to differentiate it from a similar linkage named "Article's Coauthor" between the same forms. A multiword name must be enclosed in quotes when used in a look-up formula. Because each linkage actually defines two look-up relations (one in each direction), names may be assigned to both.

Several relationships can exist simultaneously between two files in DataEase. Relationships also can be defined within a file—for example, the look-up of an employee's manager within the same file of employees. The ability to review and modify predefined relationships makes it easy for the application designer to keep track of linkages between files.

Ad hoc relationships are used in the report query process and are more powerful than predefined relationships. They cannot be used for data validation or look-up during data entry. In the DataEase Query Language (DQL), they are used widely for data selection. The term *relationship value* refers to the set of values selected from those produced by the operation of a *relationship operator* on a relationship. The syntax of the relationship value is:

<relationship operator> <relationship> <value>

In DQL, a relationship value can be used anywhere a field name can be used. Comparisons are not restricted to the equal sign as in predefined relationships. Relationship operators may be used on predefined as well as ad hoc relationships. As an example, a relationship value may be used to select all sales representatives in a file of employees by examining their primary job descriptions in a related file of jobs. A predefined relationship links a primary job code in the Employees file with descriptions in the Job file. In DQL syntax, the code to select the appropriate employee records would be:

```
For Employees with
        any Primary Job description =
    "Sales Representative" ;
```

The relational operator **any** selects the first record defined by the rest of

the statement. It should be used only in many-to-one or one-to-one relationships. **Primary Job** is a predefined relationship linking the employee's job code field to the one in the Job file, using the equality operator. **Description** is the name of a field in the job file, which is compared to the constant value **"Sales Representative"**. The result is a selection condition for evaluation of records in the Employees file.

Besides **any**, valid relationship operators are: **all**, **count of**, **sum of**, **mean of**, **highest of**, and **lowest of**. The **all** operator produces a set of values from one-to-many relationships. Because it produces multiple values, it cannot be used in a formula or as part of record selection criteria. It is used in the **list** portion of a query to output the multiple values. Operations performed with **all** can be nested to retrieve values from files in chained relationships, but not below the **any** operator.

The **count of** operator counts the number of related records for each record in the primary file and returns a number. This operator is useful in one-to-many relationships and also can confirm the existence of a related record in one-to-one or many-to-one relationships. **Sum of** returns a total over a field in the related file, which then can be used in selection criteria for the primary file. The query

```
For Departments with sum of Employees
  Salary > 100,000...
```

selects all departments that have a salary budget of more than $100,000.

The results of **sum of** and **count of** can be divided to produce a mean value, but **mean of** performs the same function in one operation. The **highest of** and **lowest of** operators return the indicated values. Nesting, as in:

```
... highest of Employees count of Bonuses ...
```

selects the employee from each department or region who received the highest number of bonuses.

The ability to define ad hoc relationships within a single file can produce impressively elegant solutions to complex problems. The query

```
For Employees with ...
  . . .
  list records
    count of Employees
  named "Same City" with
        (city = Employees city) ;
    count of Employees
  named "Same Zip" with
        (zip = Employees zip) ;
  . . .
```

establishes two ad hoc relationships named **"Same City"** and **"Same Zip"** within a single Employee file and produces counts of the number of employees living in the same city and in the same zip code as each employee.

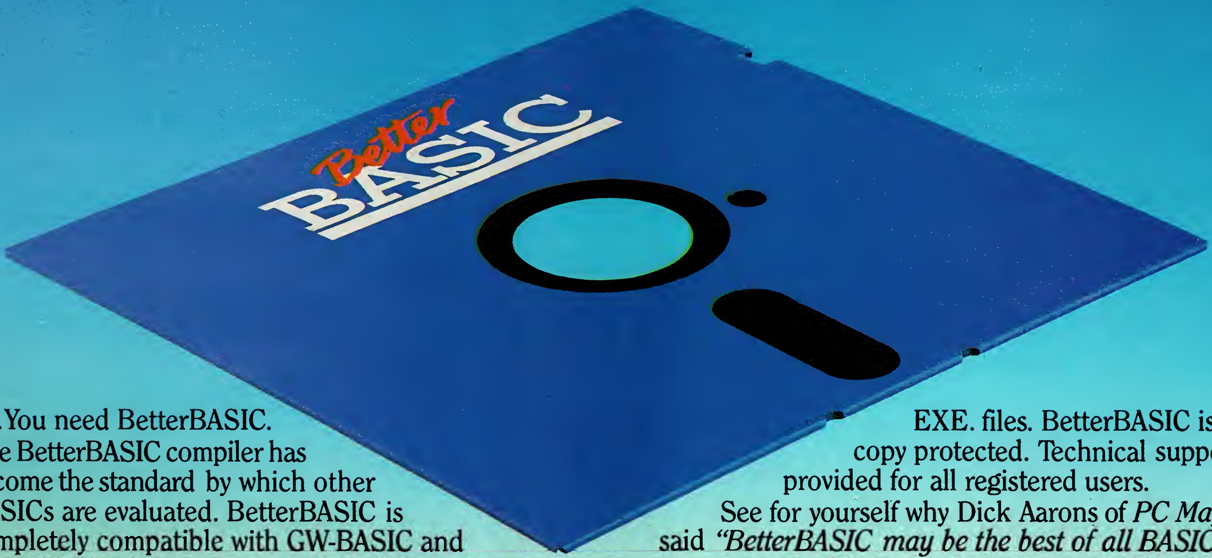
DATABASE DESIGN

The sample application used for this data manager series was implemented in DataEase, defining the three given files as forms, called Issue, Author, and Articles. Except for one report and one field data validation condition, DataEase handled the application well.

The Issue form consisted of five fields for Volume, Number, Month, Year, and Deadline. The resulting file actually contained only Volume, Number, and Deadline, because Month and Year can be calculated from Volume and Number. The Number field range check for numbers from 1 to 12 was trivial to set up using the range check feature of the DataEase field definition. Presenting the name of the month in English was also simple. In other data managers, a table, array, or string would be set up to hold the month names and used to retrieve the appropriate word. In DataEase, the predefined function **spellmonth(Number)**, used in the formula for the virtual field Month, returned the full month name when given a number from 1 to 12. The virtual field Year was presented with the formula 1982+Volume. Deadline was implemented as a date field. The only information contained in the Issue file is the deadline date, which is probably a function of the Volume and Number fields and could therefore be included in the Articles form as a derived field.

The import data files provided for the Author form had separate fields for the author's last and first names, and the application requirements specified an index on combined author name. In terms of reporting and query requirements, an index on the last name field alone was sufficient. Alternatively, the names could be combined into one field with last name followed by a comma and the first name. This approach was used to run the file loading benchmark for consistency with the testing of the other data managers in this series. Another technique would be to define a separate, derived, indexed, virtual field using text manipulation functions to determine the name from the separate Last Name and First Name fields. Because DataEase has functions named **LastFirst(name)** and **First-Last(name)**, which present a name field either way, most applications should

WHEN YOU NEED ACCESS TO FULL MEMORY, STRUCTURE, COMPATIBILITY WITH GW- & PC-BASICA, OR WHEN YOU NEED THE POWER AND FLEXIBILITY OF C OR PASCAL...



. . .You need BetterBASIC.
The BetterBASIC compiler has become the standard by which other BASICs are evaluated. BetterBASIC is completely compatible with GW-BASIC and PC-BASICA when running on IBM PC's and true clones. You can load and run your existing BASIC programs in BetterBASIC. It uses standard MicroSoft syntax and gives you more than 150 additional statements such as XREF, DEFINE WINDOW, MAKE MODULE, and PROCEDURE. In benchmark comparisons, BetterBASIC is five times faster than interpreted BASIC. There is optional 8087/80287 math chip support, and an optional Runtime System to create stand-alone EXE. files. BetterBASIC is not copy protected. Technical support is provided for all registered users.

See for yourself why Dick Aarons of *PC Magazine* said *"BetterBASIC may be the best of all BASIC programming worlds"* and selected BetterBASIC as "Editor's Choice" (Oct. 29, 1985).

BetterBASIC	$199
8087/80287 Math Chip Support	$99
Runtime System	$250
Sample Disk with Tutorial	$10

$25 REBATE — ACT NOW!
BetterBASIC is offering a $25 REBATE to its customers. Ask your participating dealer, or phone
1-800-847-7078
for a dealer near you.
Offer expires Oct. 31, 1986

Ask your dealer or call to order:
1-800-225-5800
In Canada, call 416-469-5244
Summit Software Technology, Inc.™
106 Access Rd. Norwood, MA 02062

Better BASIC™

CIRCLE NO. 195 ON READER SERVICE CARD

BetterBASIC is a registered trademark of Summit Software Technology, Inc. IBM PC, XT, AT are registered trademarks of International Business Machines Corp. Microsoft is a registered trademark of Microsoft Corp. Tandy is a registered trademark of Tandy Corp.

use a single indexed name field with the last name entered first.

Validating the two-character state abbreviations was accomplished using a Choice field. The author's work and home telephone number fields were defined as numeric strings, with the predefined phone number format. In the sample Articles form that is shown in figure 1, the formula to retrieve the author's telephone number is **lookup Author "Home Phone"**. A similar relationship could be defined to retrieve the coauthor's telephone number with the ability to define and use several relationships between two files at the same time. Even though the standard DataEase telephone format (NNN)-NNN-NNNN was used in the Author file, it was defined as a virtual numeric string with the format NNN/NNN-NNNN in the Articles file.

The SSN field also was a numeric string with the predefined Social Security Number format; it was defined to be unique. The Biography field, which was 200 characters in length, was placed outside of a box surrounding the other author information. This was necessary because DataEase does not allow the user to define points at which to wrap fields longer than a screen width. The field wraps from the right-hand edge of the screen to the left, overlaying any box or border characters.

The Articles form provided the most strenuous test of DataEase form design. This file contained look-up relations linking it to both the Issue and Author forms and had an interesting data validation exercise involving the Category and Department fields (which is discussed below).

To link the Articles form to the Author form, a relationship was established using the Author Last Name and Author First Name fields. These were separate fields in the sample application, so they were both used to ensure that the correct record was found.

ENTRY, UPDATE, INTEGRITY

DataEase is designed for data entry operations. With the extensive data validation, derivation, and look-up capabilities, many of the tedious programming chores required in other data managers are nearly automatic in DataEase. The technique of embedding data validation operations in the form design takes the place of screen and application generators in other systems. The ability to specify field-level help with automatic pop-up windowing should be greatly appreciated by designers creating systems for inexperienced end users.

Relational data managers for the PC often have difficulty updating data in related files, but DataEase handles this task elegantly. When entering data in a form, files related to that form may be called with F10-Multi. If the cursor is on a field used in a predefined relationship when this key is pressed, the form related through that field is automatically called to the screen.

In the sample application, if the cursor is in either the Author Last Name or Author First Name field, the F10-Multi key calls the author's record in the Author form to the screen. The form used to define the related file replaces the current form until the F4-Exit key is pressed. While in the related form, all data entry functions are available, including modification of the data in the key field linking the files. Suppose that while entering an article written by John Smith, the user discovers that the phone number looked up is incorrect; pressing F10-Multi brings up the author's record so the phone number can be changed. While there, the user discovers the author's last name is misspelled and changes it. F4-Exit returns the user to the current record in the Articles form, where the spelling of the author's name has to be changed manually. Of course, any other articles written by this individual need to be located and modified. Reorganizing the Articles form may indicate in the exception file where related look-up fields could not be found, but the field looked up needs to be a required field for this to occur.

If the cursor is not on a field used in a relationship when F10-Multi is pressed, DataEase presents a menu of the other forms in the database. This multiview feature is quite convenient for data entry. When entering a new article in the Articles form, if no record exists for the author in the Author file (the fact that the telephone number does not appear indicates that the author was not found), then pressing F10-Multi would bring the Author form to the screen, positioned at a new record; the name fields are filled in with the names entered on the Articles form. At this point the user can complete the author record, press F2-Enter to save it, and F4-Exit to continue with the entry of the Articles record.

For elaborate data look-up and field derivations, data entry may be slowed down by the calculations. The user can suspend the calculations by pressing Alt-F9-Suspend. Alt-F10-Derive recalculates the record on the screen, and a second Alt-F9-Suspend toggles

Imitation is the sincerest form of flattery.

But for *fast* backups, FASTBACK* is still the only choice

It seems that everyone with a floppy-based hard disk backup program is comparing themselves to **FASTBACK*** these days. Well, we're flattered.

The Industry Standard

After all, with the majority of *Fortune 500* companies *standardized* on **FASTBACK***, it's only natural for our competitors to compare themselves to us. Problem is, there's no comparison.

The *fastest* backups, period.

FASTBACK* can back up a *full* 10-megabyte hard disk in less than 8 minutes flat on standard 360K floppy diskettes! *No other program can even come close to matching our speed.* So when a competitor says they can make "average" daily backups in two minutes, they're talking about a small (1 meg) *incremental* backup that **FASTBACK*** can make in *less than a minute!* And on the IBM-AT & compatibles, we're even *faster:* **20** megabytes in 8 minutes on high-density 1.2 meg diskettes.

Simple power.

You won't find **FASTBACK*** cluttered with "file management" utilities and complex options to slow you down. **FASTBACK*** is a sleek, powerful program designed for *one purpose:* to safely back up and restore the vital data on your hard disk *as fast as possible.* Easy-to-understand prompts, interactive screen display and full wildcard naming conventions give you total flexibility to make full, partial or incremental backups quickly and easily.

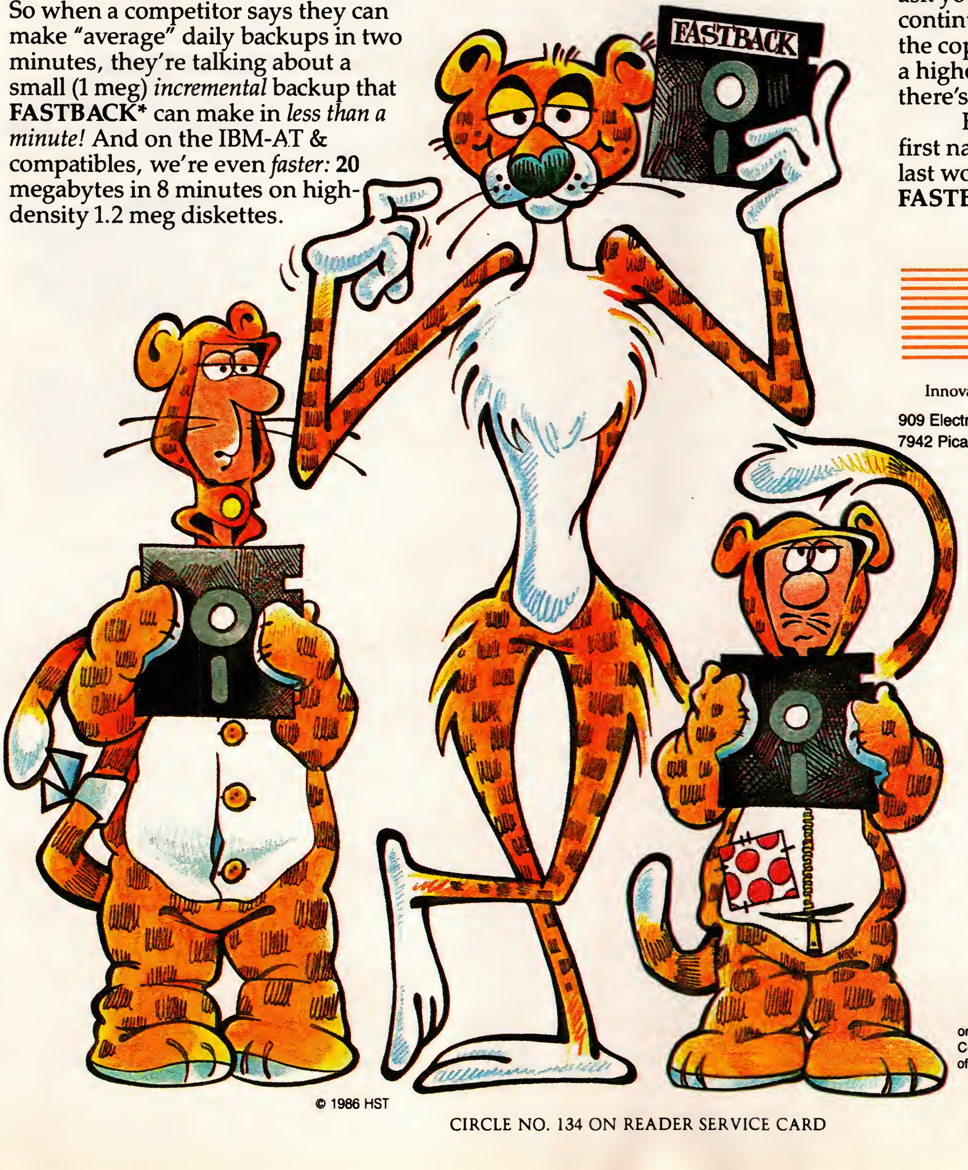
Error-correction vs. "disk mapping"

Some of our erstwhile flatterers try to make a virtue of the fact that they use standard DOS disk format for backups and "map out" bad sectors. But that won't do anything for disks damaged *after* the backups were made. **FASTBACK***'s special brilliance is a proprietary format that not only allows each disk to receive *more* data *much* faster than DOS, it also encodes Advanced Error Correction information that allows your vital data to be recovered *even if you staple, scratch or mutilate your finished backup* (tolerates up to 80 damaged sectors per disk)!

You get what you pay for.

The one thing *all* our imitators say is, they're *cheaper* than **FASTBACK***. And they are. So we ask you: since **FASTBACK*** continues to sell more copies than all the copycats put together — even at a higher price — don't you suppose there's a reason?

Find out for yourself why the first name in fast backups is still the last word on the subject. Get **FASTBACK*** today.



© 1986 HST

Fifth Generation SYSTEMS

Innovative Products Using Today's Technology

909 Electric Avenue, Suite 308, Seal Beach, CA 90740
7942 Picardy Avenue #B-350, Baton Rouge, LA 70809
1-800-225-2775 or (213) 493-4483

*This product is in no way associated with or has origination in Fisher Scientific or Allied Corporation. IBM-AT is a registered trademark of International Business Machines Corporation.

CIRCLE NO. 134 ON READER SERVICE CARD

automatic calculations back on again. Calculations automatically are performed before the record is saved, even if they are suspended.

The look-up function between related files is performed continuously. As soon as a field that is part of the relationship linkage is entered, the look-ups and derivations are performed and the affected fields filled in. Other data managers wait until the record is ready to be saved or until a specific command is given in a data entry screen program. This feature does not restrict the full-screen operation of the cursor, so the user is able to backtrack in order to correct entries based on feedback from the look-up operations.

Default values are easily used to speed data entry of batch information. They are provided in three ways.

- For those values that should not change over the life of the form, the defaults can be specified in the field definitions when designing the form.
- Values from the previously entered record can be used.
- The user can set up a default record by filling the form with the desired default values and then pressing Shift-F2-Enter Default to save it. The Shift-F5-Default Form key recalls the default record fields into the current record being entered, and Shift-F6-Default Field recalls the default value for the current field. A Shift-F5-Default Form, followed by changes to default data and a Shift-F2-Enter Default, modifies the default record at any time during data entry operations.

Data update is as simple as data entry. Records may be located by specifying all of or a portion of (with leading and/or trailing asterisks) any fields and pressing F3-View. Alt-F3-Continue View continues with the next record meeting the selection criteria. The asterisk (for multiple character positions) and question mark (for single character positions) function as wild cards for character matches in fields other than Number or Choice. Only the question mark is valid in date fields. Data selection criteria also can be entered into protected fields by pressing Alt-F5-Unchecked to enter the unchecked mode. This mode remains in effect only until the form is cleared, default values are retrieved, or the F3-View key is pressed to return to normal view mode.

Data integrity can be built into an application by using DataEase's substantial data validation techniques, as well as its protected field mode and security password assignments. However, users should be careful not to add a record

inadvertently; this is easily done during data update operations by pressing F2-Enter, which writes a new record, instead of F8-Modify, which modifies the record being edited. If records contain fields designated as unique, such as Social Security Number, DataEase will not allow a second copy or version of the record to be added to the file.

QUERY LANGUAGE
DataEase does not contain a structured language in the same sense as other data managers. Specific commands to accept data from the user, display and edit data, interface with the user for program logic control, or communicate with external programs are not available. Structured procedural constructs are offered in the data query and reporting language of the Full Reports facility, an extension of the Quick Reports feature that reaches beyond the specific purpose of defining and producing reports (see discussion below). The procedural language statements add a dimension to the nonprocedural DataEase Query Language and provide greater flexibility in report definition and production.

A substantial selection of functions is available for use in the query language, as well as in form field definitions. A general IF(condition, true value, false value) statement is provided, with the value of the function taking one of the two values in the second and third parameters depending on the evaluation of the condition. A problem developed with this function in the sample application when it was used in the Articles form definition for validation of the Category and Department fields. The Category field was restricted to one of the following values—Product Review, Technical Article, Department, or Tech Notebook—and was a required field. This was easily specified in the field definition. If the Category field was Department, then the Department field was required to be one of the following values: Programming Practices, Directions, Legal Brief, or Product of the Month. Otherwise, the Department field was required to be blank. Although a relatively straightforward data validation condition, some trouble was experienced when implementing it in the DataEase Articles form.

The Department field was defined as a Choice field, with the valid choices listed above. To restrict entry when the Category field was not Department, IF formulas were defined for the lower and upper range check functions, as suggested by Software Solutions tech-

nical support personnel. The first formula tried for the lower range was

IF (Category = "Department","Programming Practices",BLANK)

A similar function was defined for the upper range:

IF (Category = "Department","Product of the Month",BLANK).

"Programming Practices" is the lower range because it is the first of the four valid choices for the Category field; "Product of the Month" is the last of the four choices so it becomes the upper range. BLANK is a DataEase predefined value that matches empty values of all types and is used to test for empty fields. It also can be assigned to a field to make it empty.

DataEase rejected this, requesting a value be entered where the term "Department" was used. Technical support suggested removing the quotes because "Department" is a choice defined for the field. This appeared to work, although subsequent operations indicated that DataEase was now using the Department field instead of the constant "Department" in the comparison with the Category field. Although not logical, reversing the comparison order seemed to satisfy the syntax requirements, so the formulas were now:

IF ("Department" = Category,"Programming Practices",BLANK)
IF ("Department" = Category,"Product of the Month",BLANK)

This technique worked and prevented data entry when the Category field was not "Department." However, it did not prevent the operator from leaving the field blank when Category was "Department." Making the Department field a required field did not work either, because a required field is, by definition, not BLANK.

A further attempt was made by defining a fifth choice for the field as BLANK, but this still did not satisfy the requirement for an entry into the field. A last attempt, which should have worked, was to make the fifth choice "N/A" and either to force this value into the field through the range check formula or to allow the operator to enter "N/A" if Category was not "Department." This did not work, however, because DataEase would not accept any value other than BLANK for the third parameter of the IF function. An error message that stated, "A value is required" was issued, and the cursor was placed at the beginning of the third parameter position.

Similar messages requiring a value at certain points in query code also seemed to be inappropriate, given that an acceptable value is defined in the manual as one of the following: Field Name; Relationship Value; Constant; Blank Value; Miscellaneous Value such as page number, Temporary Variable, etc.; Function; or Formula. Other functions provide the number of the month, day, year, day of week, day of year, or week of year from a date value parameter. Similar functions produce the numeric hour, minute, seconds, or the time in HH:MM:SS format followed by AM or PM from a time value parameter.

Five text functions are in the "spell-out" category. These provide the text equivalent of the parameter for month, day of week, date, number, or dollar amount. They are quite useful and would require substantial programming to duplicate. The Spellcurrency function accepts a parameter up to 999 trillion and turns it into words, such as "Two Hundred and Forty Dollars and Fifty Five Cents."

Ten general text string functions are provided. Firstc, Lastc, and Midc extract a given number of characters from the left, right, or middle of a string parameter. Similar functions Firstw, Lastw, and Midw do the same for extracting words from text. A Length function returns the string length. Jointext joins two strings into one, but strips trailing blanks, which cannot be overridden. Given two separate fields for first and last names, an attempt was made to create a single name field with the last name followed by a comma, a space, and the first name. No combination of Jointext functions would leave the space after the comma. This is unfortunate, because the remaining two functions, LastFirst and FirstLast are designed to present name fields in the sequences Lastname, Firstname and Firstname Lastname, with the comma included where appropriate. These functions worked well on name fields entered with a space or comma between the names, but could not fully overcome the problem caused by the Jointext function. (Because DataEase is designed for single name fields, the solution may be to create one field containing last and first names. Separate virtual fields could be created for either last or first names.)

Financial functions are based on a model of five parameters: PresentValue, Rate, FutureValue, Periods, and Installment. Given any four parameters, one of the functions will return the fifth. Exponential and power functions are Exp (natural power), Log (natural log), Log10, Power (one number to the power of another), and Sqrt. Arithmetic functions are Abs, Ceil, Floor, Mod, and Random. Trigonometric functions are Sin, Cos, Tan, Asin, Acos, Atan, Sinh, Cosh, and Tanh.

An example approach to producing one of the reports required in the sample application helps show the use of the nonprocedural and procedural portions of DQL. The report to be produced consists of mailing lists of authors and coauthors who have written articles for a selected issue. Duplicate mailing labels for any one individual must be avoided, even if the person is an author of more than one article. The development of a nonprocedural query to produce this report was unacceptably time consuming.

The example query presented in figure 2 uses both nonprocedural and procedural sections and produces a temporary file of author names maintained such that duplicate records are not entered. This method is efficient, but could not be used in a production system because of what might be considered a design oversight in DataEase. The problem is that, while records may be deleted from and added to files under program control, DataEase has no command to "pack" or "zap" a file. In the example, the temporary file will continue to grow until it is reorganized by the operator. The reorganization process is initiated from the forms definition menu, but the selection cannot be made under program control. An improvement would be to add a **zap**, **pack**, or **reorganize form** command to the language or to implement a macro capability that allows a program to execute DataEase menu choices. The latter suggestion, if implemented so that parameters could be specified, perhaps through a script macro or file, would substantially extend the functionality of DataEase for the developer. Perhaps this function could be performed automatically with the menu system, by offering a selection that executes COMMAND.COM and erases the temporary file after the report is completed.

A data entry form is defined as a part of this query to collect the user report selection criteria. In this case, a simple form requesting Volume and Number is designed. A more elaborate report could ask for beginning and ending issues to define a set of issues to be processed. The data entry form is designed exactly the same as any other form in DataEase—so that elaborate data validation and look-ups can be performed if necessary.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

First Class Permit No. 66 Boulder, CO

Postage will be paid by addressee

PC
MAGAZINE
P.O. 2443
Boulder, Colorado 80321

FIGURE 2: *Sample Query*

```
Delete records in Temp Author .
For Articles with Volume = data-entry Volume and
         ( Number = data-entry Number ) ;
  IF count of Temp Author Check = 0 THEN
    enter a record in Temp Author
    Last Name := Articles Author Last Name ;
    First Name := Articles Author First Name .
  END
  IF Co-author Last Name not = BLANK THEN
    IF count of Temp Coauthor Check = 0 THEN
      enter a record in Temp Author
      Last Name := Articles Coauthor Last Name ;
      First Name := Articles Coauthor First Name .
    END
  END
END
For Temp Author ;
list records
  all Author to be printed named "Print it"
    First Name ;
    all Print it Last Name ;
    all Print it Address ;
    all Print it City ;
    all Print it State ;
    all Print it Zip .
END
```

A query was used to create a mailing list of authors in one issue. Using non-procedural and procedural sections, it produces a temporary file of author names with no duplicates.

In the example query in figure 2, the first statement deletes all records from the Temp Author file. Even though the word **all** would seem the natural choice to specify all records in the file, the term cannot be used because it is a special relationship operator in DQL. If a portion of a file is to be deleted, a statement such as "Delete records in Transactions with (Date < 01/01/86 and Posted = yes)" has to be used.

The next statement, "For Articles with...," defines the set of records to be processed in the Articles file. The input from the query data entry form is used to set the selection criteria to the issue desired by the operator at report run-time. The first IF statement uses a relationship operator to count the number of records contained in the predefined relationship between the Articles file and the Temp Author file. This relationship is defined as having the author last name and first name fields being equal between the files. A count of 0 indicates that the author of the article being considered has not yet been entered into the temporary file. If the first and last name fields had been combined into one Name field, and the Name field in the temporary file defined to be unique, the need for this check would have been removed. When the count is 0 a statement is executed that enters a new record in the Temp Author file and copies the name fields from the Articles file. The nested IF statements do the same for the coauthor.

The "FOR...END" statement is a loop that processes records from the Articles file satisfying the selection criteria. Next, the records in the Temp Author file are processed. Another predefined relationship called "Author to be printed," was previously established to relate the names in the Temp Author file into the Author file. This relationship is renamed "Print it" in the example to illustrate the flexibility of the language. This temporary renaming is useful when several ad hoc relationship operators are used in a query. The **all** operator specifies which records in the relationship are to be processed. In this case, **any** could have been used because the Author file contains only one record for each author. **All** could be used to list the set of authors who live in a particular state. The other fields needed for the mailing label are also retrieved and listed, completing the process.

The same report can be generated without using the procedural statements and the temporary file, but at a great expense of efficiency. To avoid the duplication of output names, the 900-record Author file must be completely processed one time. Each name in the Articles file must be examined to see if it appears as an author or a coauthor of one of the articles in the selected issue. Because the selection criteria involves several fields from the files (First Name and Last Name from both, and Volume and Number from Articles), DataEase ends up looking through the Articles file repeatedly. Several attempts were made to define a query that would produce this report more efficiently, but none was successful.

Queries must be entered through the DataEase program. The query files are maintained in a tokenized or encrypted form that normal text editors are not able to read. Two modes of entry are provided: interactive and non-interactive. In the interactive mode, DataEase provides syntax checking at each keystroke and displays menus at each step. One of two levels may be selected within the interactive mode. At the low level, only the more common of the statements are displayed on the menus, whereas the high level brings all legal statements into the menus. The low level also corresponds to the level of queries that may be defined in the Quick Reports function.

The interactive mode provides two benefits: choices that are legal in the current context are presented; and field names, statements, and relation names

QNX: With any other OS, your personal computer is asleep at the switch.

If real-time performance is the key to the next generation of small systems, task-switching is the key to real-time performance.

QNX task-switching has been measured using standard Intel benchmarks. Here are the results:

OS	Computer	Processor	Task Switches/ Sec
QNX™	IBM AT	8Mhz 80286	2800
QNX™	IBM PC	5Mhz 8088	787
XENIX™	Intel-286	5Mhz 80286	203
UNIX™	CODATA	8Mhz 68000	187
XENIX™	ALTOS	5Mhz 8086	96
UNIX™	FORTUNE	6Mhz 68000	95

The margin by which QNX outperforms UNIX-based systems is not accidental. QNX architecture is unique among multi-tasking small computer operating systems because it is modular, not monolithic. On the PC, this distinction is decisive: UNIX system overhead and processing demands sap any computer smaller than an AT.

Because QNX was developed specifically for smaller computers, not "kluged" from an OS written for larger computers, its structural advantages are apparent. QNX was the first multi-tasking, multi-user OS for the IBM PC (1982), the first again for the AT (1984) and the first networking OS for the PC and the AT (1984). For the forseeable future, QNX will remain the only real-time multi-user, multi-tasking OS for the PC, the AT and compatibles.

As a networking OS, only QNX provides integrated performance at the fundamental level of inter-task communications. The tasks need not be on the same machine: the messaging is network-wide.

A task responsible for handling an A/D converter on one machine can provide data and accept requests from tasks on any other machine. For this reason only QNX can implement true distributed processing, distributed file systems and distributed devices, with no need for a central file server.

Since QNX is modular, programmers can easily adapt and enhance the OS itself by adding their own system task. PC-DOS 2.1 or 3.1™ can run as one of QNX's many tasks.

With over 20,000 QNX systems installed worldwide, QNX is the undisputed standard for real-time microprocessor applications.

Multi-User	10 serial terminals per PC, AT.	**C Compiler**	Standard Kernighan and Ritchie.
Multi-Tasking	40 (64) tasks per PC (AT).	**Flexibility**	Single PC, networked PC's, Single PC with terminals, Networked PC's with terminals. No central servers. Full sharing of disks, devices and CPU's.
Networking	2.5 Megabit token ring. 255 PC's and/or AT's per network. 10,000 tasks per network. thousands of users per network.		
Real Time	2800 task switches/sec (AT).	**PC-DOS**	PC-DOS runs as a QNX task.
Message Passing	Fast intertask communication between tasks on any machine.	**Cost**	From US $450. Runtime pricing available.

For further information or a free demonstration diskette, please telephone **(613) 726-1893.**

The only multi-user, multi-tasking networking operating system

QNX for the IBM PC, AT and compatibles.

By Quantum.

XENIX is a registered trademark of the Microsoft Corporation. Unix is a registered trademark of AT&T Bell Labs. IBM PC, AT, XT and PC DOS are trademarks of IBM Corp.

Quantum Software Systems Ltd., Moodie Drive High Tech Park, 215 Stafford Road, Unit 104, Ottawa, Ontario, Canada K2H 9C1

CIRCLE NO. 239 ON READER SERVICE CARD

may be chosen from menus to avoid typographical errors.

Moving the cursor back or up causes DataEase to move from interactive into noninteractive operation. In the noninteractive mode, the user enters the query unchecked. Pressing F1-Interactive with the cursor at some point in the query code causes DataEase to perform syntax checking to that point, ignore the remainder of the query beyond the cursor, and continue in interactive mode.

Within a query, valid statements include the following:

LIST RECORDS [IN Relationship] <List-items> . — Presents records currently selected or from a relationship to the formatting and output sections of the report.

MODIFY RECORDS [IN Relationship] <Modify-items> . — Modifies either currently selected records or those that are in a relationship.

DELETE RECORDS [IN Relationship] . — Removes currently selected records or those in a relationship.

ENTER A RECORD IN <File-name> <enter-items> . — Enters a record in a file and places data in fields in the inserted record.

IF/THEN/[ELSE]/END — Typical structured code construct.

WHILE/DO/END — Typical structured code construct.

FOR <File-name> [condition;]/END — Loop to process selected records from a particular file.

BREAK . — Breaks from the FOR or WHILE loop to the next higher level.

EXIT . —Exits the query.

DEFINE "Variable" TYPE <length> . — Defines a temporary variable for use in the query. TYPE may be any of the types used as fields in forms (Text, Numeric String, Number, Date, Time, Dollar, Yes/No, Choice).

[ASSIGN] Variable := Value . — Assigns a value to a temporary variable or to a data entry form field. The key word ASSIGN is optional.

REPORTS AND QUERIES

The report production capabilities of DataEase are extremely flexible and powerful, yet as straightforward and logical as possible for such a complex topic. As mentioned earlier, DataEase separates reporting into two levels, Quick Reports and Full Reports. The same format and style definitions from Quick Reports are used in Full Reports, so formatting of output remains simple. The data entry and query sections are more powerful in Full Reports, and more complex calculations can be performed within the query section for output. Full Reports include the structured programming language and procedural and nonprocedural statements used to develop transaction and batch posting applications.

When requested to load a Quick Report, DataEase asks if it should be converted to a Full Report. The converted report then can be edited just like any other Full Report. The conversion process does not change secondary form selection criteria used in Quick Reports, so caution is appropriate.

DataEase helps prevent inadvertent overwriting of existing reports by asking if a modified report should be saved under a different name. One annoying feature of the Full Report query editing procedure is the restriction that a query may not be saved unless it is syntactically correct.

Reports consist of four parts: the user's selection parameters, or data entry; the query; the format for output; and the print style. These basic elements are common to both Quick and Full reports.

With the exception of simple filtering of the records in a form with field selection criteria when in data entry mode, queries are executed in conjunction with reports. A default format is provided for screen output from queries, so the user need only specify the selection criteria and desired output fields. A columnar report is the default output style, but other forms are easily selected, including the data entry form format. Screen output is automatically paused at the end of each screen until the user presses the Space Bar, and the report may be aborted at any time to modify the query.

The Quick Reports capability is accessed from the main menu. After selecting a form, the form's data screen is displayed. The F9-Report function key brings up the Quick Report menu from which reports may be retrieved, executed, saved, and deleted.

The first step in creating a report is to define the records to be selected. The form record screen is again presented, and the selection criteria are entered into the field areas of the record. At each field to be used for selection, the criteria formula is entered. The field area automatically expands up to 500 characters wide to support detailed and elaborate selection formulas. The wild card characters * and ? can be used in all field types except Number or Choice, and all appropriate functions from the query language also can be used for selection, including look-ups

and calculations on data from related files. For example, the count of selected employees in a different file can be used for comparison in the primary reporting file. Virtual fields can be used for selection, and the unchecked mode allows data to be entered into protected fields for the selection definitions.

Once the selection criteria are defined, the fields to be listed can be specified. The form record screen is presented, and the fields to be listed are marked with a number to designate their sequence. Additional parameters are entered to specify if each field is to be ordered (sorted) in sequence or in reverse, or if it is to be grouped. Statistics such as sum, mean, max, or min can be selected, data from predefined relationships extracted, and fields from any number of related files listed, as long as the relationship has been defined before the report is run.

The format for the output now can be selected. Standard formats include columnar, field per line, export, special, record entry, and template. For columnar reports, DataEase provides field headings, which may be modified. The field-per-line format places each field on a separate line with the field name. Export format brings up another menu of predefined formats including Lotus 1-2-3, DIF, Multimate, Mail-merge, Variable Length, and Fixed Length. The special report format permits the user to define the layout, placing fields and text where desired on an empty screen. Fields are selected from a menu of the ones that were defined in the previous step. The output field length can be modified and leading/trailing spaces can be removed. This allows the easy generation of lines such as "City, ST ZIP" with attractive spacing. The record entry format generates a copy of the data entry form, including boxes, text, and other layout characters. This format reports on one record per page. The template option allows the use of a previously defined format file.

After selecting the report format, the user can modify it by moving fields or editing text. Report formats may be up to 4,000 columns wide; DataEase permits scrolling of the screen to view areas beyond the edge, instead of wrapping the lines at the screen boundaries. Special commands define areas of the report as headers, trailers, or items. These commands begin with a period in column 1. Basic formatting includes the **.item** and **.end** pair of commands that establish the area for detail items from each record. Other commands are for page and group formatting, such as page headers and footers, and group headers and trailers. Titles and statistics such as group totals may be positioned in the group headers and trailers.

The **.item** command may include the parameter "across n wide m," which causes *n* records to be printed across the page in columns *m* characters wide. This is an easy way to define multi-up mailing labels. DataEase can suppress blank address lines in mailing labels without destroying the fixed label format. This is done by selecting "suppress spaces" for all items on the record output line, so that any totally blank lines are moved to the end of the item area for that record.

After defining the report format, the print style may be specified. On the print style form, the page layout is defined with page size, margins, type styles, lines per inch, and header and trailer positions. The output may be directed to the screen, to a disk file, or to a printer. Printers are defined on printer definition forms in the system administration tables, and printer control sequences can be used selectively within the report format.

Each portion of the report, from the data selection to the print style, can be modified independently. The report can be saved and automatically added to the menu of available reports for future execution or modification.

Transaction-oriented systems may be developed by using the data entry form with the option of automatically rerunning a report, along with the DELETE, MODIFY, and ENTER statements. The transaction data entry screen is defined as a report data entry screen. Upon the operator's completion of data entry, the query section of the report processes the transaction, modifying, deleting, or entering records in files as necessary. The data entry form is presented to the operator again for the next transaction, and the defined report is run again. An exit condition must be defined and code written to check for it so the user can leave the query.

Reports also can be designed to post transactions from one file to another. Tools are available to allow code that will validate a transaction before posting, mark the transaction as posted, modify the files, and produce an audit file entry if necessary.

The Quick Reports function was not up to handling one complex report required for the sample application. Among the requirements of the report were page numbering and date/time stamping on each page. These values are not available in Quick Reports, so

Can you pass this simple true or false test on structured programming?

1 True BASIC™ is a true, structured language.

☑ **True.** ☐ **False.**

True BASIC offers a full selection of control structures like SELECT CASE, nested IF-THEN-ELSE IF, and DO-LOOP, and external procedures which can be compiled into libraries, making True BASIC faster, easier, and more flexible.

2 True BASIC has a wide range of powerful features.

☑ **True.** ☐ **False.**

True BASIC has a complete matrix algebra package and the best graphics ever in a higher level language. And there are optional libraries for things like sorting and searching and 3-D graphics.

3 Programs in True BASIC are fully portable.

☑ **True.** ☐ **False.**

Programs written in True BASIC will run on any computer which runs True BASIC, good news for anyone who uses more than one kind of personal computer.

4 *Classroom Computer Learning* named True BASIC one of the "Outstanding Products of 1985."

☑ **True.** ☐ **False.**

Not only that, but *Byte* magazine called it "Superior to Microsoft™ BASIC." *PC* magazine said it was "the easiest to learn of all the BASICS. . . ." And finally, *Electronic Learning* concluded "Good graphics have never been easier. . . ."

100%

5 Kemeny and Kurtz can also help teach students mathematics.

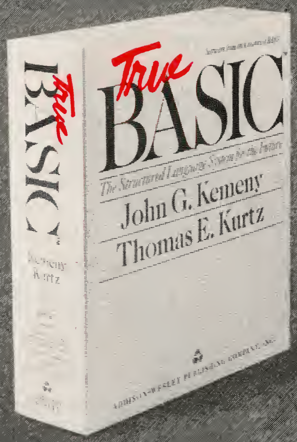
☑ **True.** ☐ **False.**

True–with the Kemeny and Kurtz Mathematics Series, including programs on algebra, trigonometry, pre-calculus, calculus, probability theory, statistics, and discrete mathematics. They're reasonably priced at only $49.95 per single copy with quantity discounts available.

O.K. You've made me a True believer.

Whether you're programming for your own applications, teaching others, or developing products to go to market, you should consider the advantages of True BASIC. After all, the true test of a good product is adoption. And True BASIC has passed that test. High schools, universities, and corporations around the world have chosen it. It's a flexible, powerful structured programming language that you can depend on, 100 percent. True BASIC is available for the IBM™ PC and compatibles, Apple Macintosh,™ and Commodore Amiga.™

To order call 1-800-TR-BASIC. In New Hampshire, call **(603) 643-3882.** Or send the coupon below.



True BASIC™ inc.
39 South Main Street
Hanover, NH 03755

BASIC information necessary to pass this test: John Kemeny and Tom Kurtz invented the original BASIC programming language. Now they're back with an even better version: a flexible, easy-to-use structured programming language they call True BASIC.™

System requirements: IBM PC 192K, DOS 2.0 +; Macintosh 128K; 512K recommended. Amiga 512K

CIRCLE NO. 191 ON READER SERVICE CARD

30 Day, No-Risk Trial

Act now to take advantage of this special offer. Order today. The program costs only $149.90. Use it for 30 days, and if you're not completely satisfied, send it back for a refund.

I want to order the True BASIC Language.

My computer is: ☐ IBM PC/compatible. ☐ Apple Macintosh. ☐ Send me True BASIC product information. ☐ Commodore Amiga.

Name ______

Title ______

Company/University ______

Address ______

City, State, ZIP ______

Telephone ______

Credit Card, Number, Expiration Date ______

PCT-9

the report was automatically converted to a Full Report and modifications to the query made to obtain access to the values of current date, current time, and current page. With this one exception, Quick Reports could have been used for the entire report, which listed article titles, author names, and editorial and listing page counts for each issue, sorted on volume and number; it also gave subtotals of each page count type and a grand total for each issue. Many data managers would benefit from a report writer with as much power and simplicity as DataEase's Quick Reports.

SORTING AND INDEXING

DataEase provides indexing on single fields only rather than on a formula combining multiple fields. Indexing is permitted on virtual and derived fields, and multifield indexes may be simulated in this manner. A separate B-tree file is created for each field indexed, with the hexadecimal representation of the field number embedded in the file name extension. The approximate size of an index file is [1,024 + (2 * Number of records * indexed field size)].

DataEase contains no SORT verb, but sorts are automatic when required. They are initiated when a report definition requests output to be ordered, except when only one order is requested and the field to be ordered is indexed. Sorts that are too large for available memory will abort. The documentation discusses this in detail and provides work-around suggestions.

The amount of memory required for a sort is calculated by the formula:

```
( 2 * Number of records in the report)
  + (( 7 + Average size of sort field )
    * Number of records to be sorted )
```

For a primary sort (that is, a report ordered on a single, unindexed field), the number of records to be sorted is the same as the number of records in the report, which may be different from the number of records in the file, depending upon selection criteria. Consider the following two examples from the DataEase manual:

```
... (query selection criteria) ...
list records
     Name in order ;
     Address .
```

If the name field is not indexed and averages 20 letters in size including embedded spaces, and 10,000 records have been selected to be sorted, then the memory required is (2 * 10,000) + ((7 + 20) * 10,000) = 290,000 bytes, or 29 bytes per record.

For a multilevel report, the memory required for a secondary sort is determined by the number of records in the largest group resulting from the primary sort. Ordering of levels below the primary level is always done by sorting, even though the field may be indexed.

```
... (selection criteria) ...
list records

     Department Code in order ;
          Zip Code in order ;
          Name ;
          Address .
```

If the Department Code field is indexed, then the number of records that can be ordered by Department Code is unlimited. For each department group, a sort will be performed on the five-digit Zip Code field, regardless of any existing index on Zip Code. If the largest department has 10,000 employees, and the selection criteria produces 40,000 records for the report, the memory required is (2 * 40,000) + ((7 + 5) * 10,000) = 200,000 bytes.

A secondary sort that requires too much memory often is simulated by

forcing a pass through the file for each group of the primary field and using an index on the offending field. If the above report cannot be produced as stated, an index on Zip Code could be created and a pass made through the file for each department code.

With a 640KB machine, memory restrictions for sorting should not occur except on very large files. The 230KB DataEase program uses remaining available memory (after DOS and resident programs) as a data area. This could be up to 380KB on a 640KB machine if DOS and resident programs do not occupy more than 32KB. The DataEase data area is used for storing records and forms relating to current database activities, but a substantial portion of it is available for sorting.

The sort sequence order can be specified as "in order" or "in reverse." The collating sequence depends on the field type. Text fields sorted "in order" list numbers first, letters alphabetically, and punctuation marks in ASCII sequence. Number, Numeric String, Date, and Time fields are sorted in their natural orders. For Yes or No fields, No is listed first. Choice fields are sorted in the order of the choice number, not the definition text. Sorts on the definition text can be accomplished by assigning the field to a temporary variable in the report definition and specifying the temporary variable as the field to be listed in order.

The report phrases "in groups" and "in groups with group-totals" initiate sorting, but also permit group headers and trailers, statistics, and suppression of repeated field lines.

IMPORT/EXPORT

A data import facility is provided as an option within the DataEase utilities menu. An import process may be defined and executed as a one-time import, or the definition may be saved for reuse as necessary. Import definitions may be modified, deleted, or executed from the import menu. Data export is handled within the reporting functions, and the export format is defined in the report formatting phase.

DataEase exports to Lotus 1-2-3, DIF, Multimate, Mail-merge, Variable Length, and Fixed Length formats. The standard output formats for variable- and fixed-length export can be modified to specify field lengths, positions, and delimiters.

DataEase imports from other DataEase databases, Lotus 1-2-3, Symphony, DIF, dBASE II and III, Mail-merge, Variable Length, and Fixed Length. The fields to be imported may be specified, and data imported may be used to update and/or add to existing data in the destination form.

For selective field transfer, DataEase expects the import file to be organized by field name and uses these field names to match field names in the DataEase destination form. For DataEase and dBASE formats, the import source field names are automatically extracted from the data file header. For other formats, the first record of the import file must contain the field names. For variable-length import files, such as delimited files from other programs, the field name record may be added by the program producing the file, or they may be added with the DOS COPY command. Providing DataEase with the name of a separate file containing the field names would avoid this step. Selective field transfer cannot be performed with fixed-length files.

If the import fields match the destination form in number and sequence, all fields may be transferred without using the field name header record.

Considerable flexibility is provided for matching import fields to DataEase fields. The delimiters between fields and records in variable-length files may be specified, and DataEase can be directed to insert decimal points in fixed-point number fields if the source file does not contain them. For text fields, leading and trailing spaces and surrounding quotes are stripped for DIF and variable-length imports. Only digits for numeric strings and time fields are imported; punctuation characters are ignored. Dates may be in a six-digit format and contain slashes. Slashes are required if two digits are not used for each field, as in 5/9/86. For number and dollar fields, all digits and the plus or minus sign are accepted. The sign may be at either end of the field. Scientific notation is also acceptable. Choice fields accept either the number of the choice, if the import field starts with a digit, or the choice definition text.

For all formats, DataEase can be directed to check for duplicate records and told what to do when they are found. A record is duplicate if it matches all the UNIQUE fields in the destination form. The options are: "Add non-matching," "Update matching," "Add or update," and "Do not match." The "Add non-matching" option adds new records to a form only and discards duplicate records. The "Update matching" option uses import records only to update records already in the DataEase form and discards new records. The "Add or update" option uses import records to update existing records when duplicates are found and adds non-matching records into the file. "Do not match" adds records to an empty file or to one in which it is known that no duplicates will be identified. This option saves the time that would be spent checking for duplicates.

During data import, all derived fields whose values have not been imported are derived, and all derived fields are rederived when an import record updates an existing record. Each record is checked for legitimate field values as specified in the form definition, and errors are logged by record and field number to an exception file that may be either examined with a text editor or printed.

Once data import has started, the process cannot be interrupted except by rebooting the computer, so trial imports on sample data sets should be performed before attempting to import large files for the first time.

A helpful feature of DataEase is its ability to recover data after a system crash or disk errors. Detailed directions are provided in an appendix to help recover from several possible difficulties, including lost or damaged files and inconsistencies caused by insufficient disk space or aborted import operations. Sufficient information is provided about the layout of data files so that a damaged file could be repaired through direct patching with DEBUG or a disk utility. DataEase documentation defines the file naming conventions and data file record formats, and its form-reorganizing process recreates index files and verifies the consistency of the data. Menu choices are available to back up and restore DataEase databases.

DataEase files appear to be left in safe states whenever not in use for an operation. During the research for this article, the databases survived several power outages, equipment failures, and system resets.

Some of DataEase's operations, such as data import and form reorganization, cannot be aborted once they are under way. Any data import operations should be tested on a small file to ensure form consistency with the incoming data format before any large files are to be imported.

END-USER FACTORS

DataEase was originally designed for the end user, and only recent enhancements have been oriented toward developers. The menu structure is clear and straightforward, and the pop-up help screens, when available, usually provide an answer or a reference to the appropriate section in the manual.

The assignment of functions to the function keys seems to be as consistent as possible throughout the various functions and menus, but some inconsistencies exist in option selection procedures. In some cases, just typing a number is sufficient to select an option from a menu. Other cases require typing the number followed by Enter.

The user is frequently asked a question to be answered "1: no" or "2: yes" is offered. In most cases, the letters *n* or *y* also trigger the correct response, but sometimes this does not work, and the numeric selection must be made. To abort an operation, Esc is used in some cases, but F4-Exit must be used in others. When an error is made in a field derivation formula, DataEase beeps and moves the cursor to mark the offending position. As soon as a key is pressed, the cursor jumps back to the beginning of the formula and must be moved with the right arrow key back to the required position.

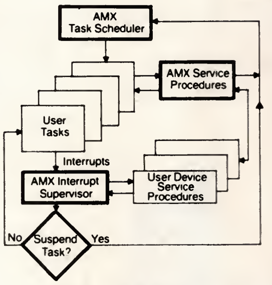
AMX

8080
Z80
6809

8086
8088
Now
68000

Real-Time Multitasking Executive

- No royalties
- Source code included
- Fault free operation
- Ideal for process control
- Timing control provided
- Low interrupt overhead
- Inter-task messages



Options:

- Resource Manager
- Buffer Manager
- Integer Math Library

AMX is TM of KADAK Products Ltd.
CP/M-80 is TM of Digital Research Corp.
IBM, PC DOS are TM of IBM Corp.

- Language Interfaces:
  C Pascal
  PL/M Fortran
- DOS File Access:
  CP/M-80
  IBM PC DOS

AMX		
for 8080	$ 800	US
8086	950	
6809	950	
68000	1600	
Manual (specify processor)	75	

KADAK Products Ltd.
(604) 734-2796
Telex: 04-55670
206–1847 W. Broadway, Vancouver, B.C., Canada V6J 1Y5

CIRCLE NO. 258 ON READER SERVICE CARD

OVERCOME

FORTRAN | PASCAL | C

LIMITATIONS WITH

$129
NO LICENSE FEE

NO LIMIT

Ver. 3.0 For:
MS FORTRAN/Pascal/C
IBM Professional
and R-M FORTRAN
SuperSoft FORTRAN

NOW SUPPORTS IBM ENHANCED GRAPHICS AND HERCULES MONOCHROME GRAPHICS

A library of over 90 Assembler routines transform FORTRAN, Pascal and C language compilers into the flexible, responsive, complete languages needed for the microcomputer environment. Hundreds of NO LIMIT owners are creating highly interactive software systems, often utilizing existing mainframe code and saving time and money in the process. Ver. 3.0 features include:

EXTENSIVE GRAPHICS	(Get, Put, Paint, Color, Dot, Line, Box, Circle, Ellipse)
FULL SCREEN CONTROL	(Windows, Cursor, Read/Write Screen)
STRING MANIPULATION	(Match, Compare, Concatenate/Extract, Pack, Justify)
KEYBOARD CONTROL	(Read Key During Execution, String Read With Edit)
FILE MANAGEMENT	(Exist?, Rename, Delete)
COMMUNICATIONS	(Full Interrupt Driven to 9600 Baud, Set Com Line, Send/Receive, Line/Modem Status)
INTERRUPTS	DOS Execution, Program Execution
OTHER FEATURES	(Command Line Read, DOS 3.1 Directories, Peek, Poke, Random Numbers, System Status, etc.)

And to complement NO LIMIT, the I/O PRO development system allows creation and editing of FORTRAN/Pascal/C callable screens. This word processor type system pays for itself several times over on your first project. $261.

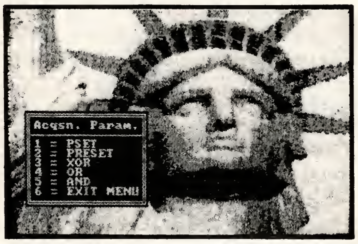
M | E | F Environmental Inc.
P.O. Box 26537 Austin, Texas 78755
(800) 562-9700 (512) 251-5543 (Texas)

©Trademark of Microsoft Inc., Supersoft Inc., IBM Corporation, Ryan-McFarland and Microsoft Inc., respectively.

CIRCLE NO. 133 ON READER SERVICE CARD

IMAGE ACE II

Video Capture System



$295.00
— complete —

- single slot card
- software disk
- complete manual
- video cable

Digitize video images from cameras, television tuners, and video recorders directly to your IBM PC display.

- No additional monitor needed
- 320 x 200 x 4 levels: B&W or pseudocolor
- Continuous, single frame, and remote acquisition modes with 1.3 second capture
- Works with IBM PC, XT, AT, or compatible with std. color/graphics adapter
- Save compressed 16K images on disk, send via modem, or print on matrix printers
- Combine images with text
- Make hard copy printouts for reports, presentations, signs, and posters
- Display captured images on almost any other PC

Applications range from production line monitoring, product inspection, machine vision, and robotics; to video telecommunications, BBS image exchanges, personnel records, and visual data bases.

Lodge Electronics
(IBM, PC, XT, and AT are trademarks of IBM Corp.)
P.O. Box 338 • Streamwood, IL 60103 • (312) 837-6553

CIRCLE NO. 127 ON READER SERVICE CARD

TURBO PROFESSIONAL™

SERVICE INTERRUPTS
No assembly required

RESIDENT PROGRAMS
Easy, pop-up routines

EXECUTIVE PROGRAMS
Run ANY DOS program

DISK SECTOR I/O
Lowest level access

FAST TEXT WINDOWS
Virtual windowing system

KEYBOARD MACROS
Simple, powerful

LOTS OF EXAMPLES
21+ full example programs

MUCH MORE. . .
Over 140 routines in all

"If you never thought Turbo Pascal was a systems programming language, you've never seen Turbo Professional."
Darryl Rubin
Computer Language

For programs that move with technology—Turbo Professional—a truly professional library of subroutines.

150 page reference manual.
Full source—many example programs.
No royalties charged for applications.

Requires IBM compatible, DOS version 2.0 or greater, Turbo Pascal 2.0 or greater.

Turbo Professional, trademark of Sunnyhill Software
Turbo Pascal, registered trademark of Borland International

Dealer Inquiries Invited.

$69.95 plus $5.00 S&H

Washington residents add 7.9%
International orders add $5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

Sunny Hill Software
13732 Midvale N. Ste. 206
Seattle, WA 98133
(206) 367-0650 M-F, 8-6 PDT

CIRCLE NO. 152 ON READER SERVICE CARD

When query code extends beyond one screen, a second page is created. However, the page break position is fixed; the display cannot be repositioned so that the last line of the first page and the first line of the second page can be viewed together.

Despite these inconveniences, DataEase's end-user interface is very good. Menus for applications are easily defined. Forms, reports, and other procedures can be organized into menu structure through the custom menu facility, so that the user does not have to use the standard menus. Menus can call other programs and execute DOS commands, and they can execute utility procedures such as predefined imports with parameters.

A chain menu can be defined to specify a series of operations to be accomplished as a single menu choice from another menu. This might include exporting data to a spreadsheet format file, calling up a spreadsheet program to process the file, and importing the modified file. Chain menus are not displayed and operate as a sophisticated batch language. A chain menu can be made into the start-up menu and assigned to a user in the system user record. This causes the automatic performance of predetermined functions when the user starts and exits DataEase. If one of the items in the chain is the top-level menu for the application, the functions in the start-up chain menu before the top-level menu call are performed automatically when the user signs on, and the functions after the top-level menu call are performed when the user exits. Each user may be assigned a different start-up menu, and each menu may have a security level.

Menus are defined and maintained as records in a system menu form. Up to nine choices can be defined on a menu. The form provides fields for choice description, function name, and function type. The 11 function types are:

Main menu — to call the DataEase main menu
User menu — to call a user menu
Record entry — to call up a form for data entry or changes
Query — to initiate an ad hoc report
Run report — to run a predefined report
Status — to display the database status
Backup and **Restore** — to save and restore the database
Utilities — to call the utilities menu
Import — to import data
Program call — to call an external program or DOS command

Parameters may be passed to external programs, and DataEase permits parameter definition to be deferred for user entry at the time of execution of the menu choice. With these features and the flexibility of menu definition, elaborate and effective menu systems can be designed.

POWERFUL ASSETS

DataEase performed well on the five benchmarks used to test data managers reviewed in this series—in the same controlled PC/AT machine configuration. Table 1 shows the results.

The benchmarks were easy to set up in DataEase. The Full Reports capability was used for the benchmarks that involved output generation or data file modification using the Author file.

In the benchmark in which each unique state code is listed in order with the count of records for that state, the query section of the report is:

```
For Author ;
List records
    State in groups ;
    State > "AA" : count .
```

CIRCLE NO. 157 ON READER SERVICE CARD

CIRCLE NO. 185 ON READER SERVICE CARD

CIRCLE NO. 172 ON READER SERVICE CARD

CIRCLE NO. 158 ON READER SERVICE CARD

TABLE 1: *Benchmark Results*

BENCHMARKS (seconds)	**DATAEASE TIME**	**AVERAGE TO DATE**
Add 900 records to an empty database table	112	183
Index table on two fields (7 bytes)	62	46
Document and tally codes from one column	47	53
Mass change of one column (28 rows of 900)	5	23
Extract selected records to create a text file	3	12

Note: All benchmarks were run on an IBM PC/AT (6 MHz) with 640KB memory. The tests were run in an 8MB partition on a CMI 20MB hard disk under DOS 3.0.

DataEase compares favorably with the 12 previously reviewed data managers. In four out of the five tests, DataEase logged better times than the average.

In the format section of the report, the output fields were specified in the group trailer to avoid printing a line for each record. The report required only a minute or two to set up.

The benchmark to change each occurrence of the state code from "CO" to "CL" used the query code:

```
For Author with State = "CO" ;
Modify records
     State := "CL" .
end
```

The last benchmark, which was to produce a delimited file of California authors in Zip Code sequence used the query code

```
For Author with State = "CA" ;
List records
...
```

followed by a list of field names. In the format section of the report, the variable-length export format was selected, and the field and record separators were designated as a comma and carriage return, respectively.

DataEase version 2.5 joins the ranks of data managers for application development with substantial assets in the areas of user interface and report generation. The multiview capability for related file data entry, virtual fields, definable context-sensitive field help, procedural language, powerful report generation capabilities, and flexible file relationships should be well received by developers. Corporate application developers in organizations with an installed base of DataEase users should be especially pleased with this upgrade.

DataEase handles a wide variety of applications commonly found in the business environment. It handled the sample application quite well. The application was a good test of many of the features of DataEase, but certainly not beyond any of its capabilities or capacities. The ability of DataEase to relate one file into more than one other at the same time allowed the application to be developed without the substantial amount of programming necessary in other data managers to implement the Articles data entry screen. Care must be taken to avoid situations such as the sample application mailing label report in which query relationships become unwieldy and prohibitively time consuming. The overall performance of DataEase should be acceptable for properly designed applications.

No multiuser capabilities are available in this version of the product. Software Solutions indicates that a multiuser version is under consideration, but no details or release date information is available at this time.

The DataEase manual packs a lot of information. The single, 350-page volume contains several useful examples and comes with a 17-page addendum that amplifies and updates several of the sections. A separate nine-lesson tutorial manual with disk is also provided, along with a 32-page *Quick Reference Guide* and a pamphlet that discusses changes and upgrades for users of previous versions.

The most important attributes of DataEase are the flexibility and power of the data entry validation features and the report definition and production facilities. Software Solution's development efforts should concentrate on enhancing the command language with functionality that is suited for the down-to-earth requirements found in real-world applications. Some attention should also be paid to the consistency of the end-user interface.

Dave Browning is vice president and co-owner of WBS and Associates, Inc., a microcomputer and custom database consulting firm in the Washington, D.C., area. He is also director of vendor relations and chairman of the database special interest group for the Capital PC User Group.

PC BRAND™: CAREFULLY CHOSEN PROGRAMMER TOOLS

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out programs themselves** if product code begins with E, T, or L through N — even if it means breaking the disk seal. Some developers do pose limits, so for products beginning with other letters, opening sealed disks constitutes acceptance. But you can at least review the manual. There's just nothing stopping your buying from PC Brand.

BRIEF *Is Anything But. A Whopper of an Editor*

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text, from keyboard or files, is housed in multiple buffers, and scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, or deletion, with as many "undo" levels as you want.

Brief has text search abilities rivaling "grep", with wildcards for matching, indifference to intervening characters, acceptance of character ranges.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own Lisp-like macro language which has structure, 32-character variable names, conditional execution, loops, and you can actually read it! Nothing like the hieroglyphs we've seen elsewhere. Bulletin board and public domain disks with macros. "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.)

Ask for:	List:	PC Brand:
U0590	$195	**Call**

HALO GRAPHICS SYSTEM *Multi-Board Graphics Library*

The premier graphics library that got the ball rolling for PC-based graphics and has grown so omnipotent that it supports over 25 graphics boards — including IBM's EGA and Nr. 9 Revolution's hi-res series — and has a multitude of mouse and printer drivers. All that in each box. Separate C versions for Lattice, M'soft, Aztez, CI86. What does Multi-Halo do? A down to the last pixel graphics library plus functions to reset drivers so distributed program can run on anything. Wonderful value for single license. Costly royalties though for redistribution. Specify: S0315 & Language. List: **$300.** We: **$219.** With Dr. Halo II, a free-standing "paint": List: **$440,** Us: **$299.**

WINDOWS for C/WINDOWS for DATA *Microsoft Windows™ and TopView™ Compatible*

CALL FOR FREE DEMO!

Windows for C™ is a library of over 80 functions to add the pizazz and practicality of window partitioning to your application. Unlimited windows, each defined in a C structure for easy reference throughout your program, can be made either to pop up or permanently overwrite the screen. Routines will scroll and highlight lists with arrow keys, will read and scroll ASCII files vertically and horizontally in windows, and even write to memory-loaded files off the screen.

Logical treatment of video attributes permits unchanged programs to run on color or monochrome. Colors of windows are set individually.

All functions are in separate modules; only those used are linked. Only buffers holding on-screen or temporarily obscured windows occupy RAM; others released dynamically. Best overall rating and fastest display in Bill Hunt's 7/85 *Tech Journal* review of five windowing products.

Windows for Data comprises all of Windows for C but takes in data through the windows as well. At the high level a single function lets you specify prompt string, field length, data type, screen location, picture, target variable, then sets lesser functions scurrying to get and process a user's input. There are utilities to get system date and time, mess with strings, create your own masks for fields.

Field options can require entry, prevent entry, permit insert or overtype, beeping on invalid or overflow keystrokes, and attachment of field-specific help messages and functions you want called to display messages or validate entries. And you decide which keys will clear a field, jump to the next or prior, quit, etc. Options diverse enough that a set of "fields" can be made to behave like a Lotus™ menu.

Specify Compiler:	List:	PC Brand:
T0100 Windows for C	$195	$149
T0150 Windows for Data	$295	$250

C-WORTHY LIBRARY *Fits Out Applications with Shipshape Interface*

FREE DEMO!

Not the usual flotilla of functions for small crafting, but a formidable battle wagon for major C engagements.

C-Worthy™ wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Proof? A single function call sets up a complete text editor in a screen window!

- High level calls pop menus and scrollable lists to the screen, restoring the background when dismissed, and branching to the chosen activity in your application.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full on-screen context-sensitive help management. Keyboard entry routines look for the help key and interrupt with paged text windows explaining what to do.
- Full error message interface sends error codes to C-Worthy which discusses problem with user. Gone is all that error-checking clutter from your core program.

Your application is nested in this powerful environment. C-Worthy's architecture uses C's pointers to functions to drive your application.

C-Worthy includes utilities to maintain text files of help and error messages. Their segregation means applications can readily be translated into foreign languages without program rewriting. Plus display routines resize for text length.

Decomposed low level functions used by the high-level calls are supplied. All machine dependency is housed in interchangable overlays loaded at run-time. Thus no recompilation needed to run on a mix of PC and MS-DOS machines.

C-Worthy hands you a consistent and intuitive interface and a revolutionary design approach. Novell found it "played a key role and accelerated development" making NetWare™ easier to use. "You owe it to yourselves to take a look."

Specify Compiler:	List:	PC Brand:
T0500	$295	$269
T0550 Network License	$495	$449

C-TREE *B-Tree File Manager, Source Code, No Royalties!*

C-tree is sturdy code that has weathered many seasons of prolonged and widespread use. It comes in C source, so you can modify it to fit a special case. No royalties provided you bind it into your binary application.

C-tree's design splits nodes to allow any number of users to access an index file simultaneously even when updates are in progress. So multi-user configurations and adaptation to networks are possible. Record-locking routines are provided for DOS 3.1/3.2, UNIX and XENIX.

Thanks to source code which does not deviate from the K&R standard, C-tree can travel. Tests in many environments prove that C-tree gives your application a ticket to anywhere.

C-tree permits any number of keys for a data file, supports duplicate keys, alphanumeric or numeric, supports files of variable record length; multiple keys in one index file, and keys of variable length. Both high level ISAM routines which handle details with minimum coding, and decomposed step-by-step functions you can access directly. It's comprehensive.

Ask for:	List:	PC Brand:
F0660	$395	$329

CURSES *Unix Style Screen Management*

Curses from Lattice™ manages the screen of the PC like Unix™ curses. Library of 84 functions and macros parallels Unix with matching parameter lists. So Unix programs are at home on the PC, and vice versa. Keeps any number of screens in memory, supports color, vast function set to get characters, wrap lines, scroll, blank lines, highlight, etc. Like Unix refreshes screen only on your command. Ask for: L0850. List: **$125**. Here: **$99**. With Source: L0860, **$250/$199**

dBC *Lattice Library Maintains dBASE Compatible Files With the Power and Speed of C*

dBC™ links C to dBASE. It creates and maintains files and their indexes which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dBC can use any files created by dBASE. Now C and dBASE can operate on the same data bases interchangeably.

That opens up the widespread culture of dBASE installations to exploitation by C programmers. Tap that market, avoid the resident dBASE language, and gain the advantages of C with this single product.

dBC's functions parallel all dBASE's file handling commands, many decomposed to give closer control. Each backed by demo source files on disk.

Use dBC for custom work for clients, or on its own. It's a complete ISAM file manager for C whether or not dBASE will be used in tandem, supports all four memory models, and can have sixteen index and data files open. Big discount to buyers of both dBASE II and III versions. Specify Lattice, Microsoft 3.x, or DeSmet.

Versions:	List:	PC Brand:
L00II For dBASE II	$250	$195
LCCII With Source	$500	$390
L0III For dBASE III	$250	$195
LCIII With Source	$500	$390

PANEL *Feature-Laden Screen Design Tool*

Writing your own screenware can blow completion dates and profits. Panel™ works with you interactively to set up foolproof screen displays and data entry forms rapidly. Output is C source code.

Not just single plane: layer your screen designs with up to ten overlapping images: Background pop-up lists, help boxes, and alternate input fields.

Panel builds in a user interface for keystroke movement within and between fields, supplies validation routines for checking user field entries. Diverse attributes may be selected for any field — size, data type, color, conversion of input to upper case; clearance of existing data when new entry is started; masks for standard formats (eg, dates); phrases which fill in when their first letter is typed; multiple-choice lists from which to choose by cursoring a highlighted bar. Fields may be multi-lined and scrolled if larger than the screen space allotted them. Specify: S0400 & Compiler. List: **$295**, Us: **$229**

WHY US?:

Latest versions of all products • Everything listed is in stock • Shipped 24 hours or sooner • No surcharge for credit card or COD purchase. **NEED TERMS?** On-the-spot credit to most public companies, government, educational, medical institutions. **LOOKING FOR SOMETHING?** We can get many more products — just ask! **NEED MORE INFO?** Our Catalog and literature cover just about everything.

For Orders, Literature, or Catalogs, Call Us at…

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

SAVE $5 Just tell us this code: **J8**

TODAY's TOP QUALITY AIDS TO PROGRAMMING PRODUCTIVITY

GREENLEAF FUNCTIONS *Bountiful Harvest*

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual helps select functions, bulletin board, too.

A sampling: *DOS* extensions for file and directory manipulation; *Screen:* to select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; pixel get and put; read light pen. *Strings:* Center, justify, etc.; efficient list operations which add, delete, sort string pointers for top speed. *Other:* graphics character primitives, keyboard status, function key assignment, time/date, read registers and memory size, peek and poke. Mature best-seller. Specify: S0770 & Compiler. List: **$185**, Here: **$139**

PFORCE *Phoenix Pfunction Pfestival*

Lotus® didn't do badly pulling it all together in one place. Phoenix has followed suit with the ultimate integrated C library, offering everything from low level functions for hardware access to complete b-tree database management. Along the way are prerequisites such as string manipulation, time/date, field and screen editing, but also four styles of menus (Lotus included), windowing, background tasking, DOS interfaces, directory management, even interrupt-driven communications. Design emphasizes objects, so characteristics of windows, databases, records and fields can be initiated and changed outside functions.

One large collection in place of bits and pieces means one set of instructions and PforCe™ has tutorials, extensive examples, quick reference, and on-line help.

Everything in source, no royalties, all memory models of Lattice, M'soft. Specify: S0220 & Compiler. List: **$475**, PCB: **$349**

GREENLEAF COMMUNICATIONS *Hello World*

Want your application to communicate with other users or remote date bases by asynchronous communications built right into your C programs! Even if you don't need it now, that's a skill to have at the ready!

120 functions *and demo programs in both C and assembler source code* set up separate transmit and receive ring buffers for up to 16 simultaneous channels. Interrupt driven so you can halt an incoming record, display it, file it, let the user edit it, then continue. Goodbye separate communications software.

Supports up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Specify: S0750 & Compiler. List: **$185**, Us: **$139**

PRE-C *Pick the Lint from Your Program*

Pre-C is like UNIXs lint. It finds problems your compiler won't. Problems that a debugger will have trouble figuring out. Even problems which will cause trouble with other compilers.

Compilers see one module at a time. Modules only meet at link time. Pre-C looks at *all modules at once* and reports conflicts in data type declarations; function call parameters which disagree with functions, machine-dependent expressions which inhibit portability. It spots obsolete usage (even C changes), casts with suspect conversions, variables never used, functions never called, unreachable code. Adheres to UNIX System III compile standard to ensure your portability. Ask for: P0590, List: **$395**, Ours: **$279**

DAN BRICKLIN'S DEMO PROGRAM

Storyboard Your Program

The Legendary One has created Metaphor Two when the rest of us are still on Zero. Dan's first was the original electronic spreadsheet (VisiCalc™). This one is for programmers.

Words don't express program ideas because programs are screens! Dan's Demo creates slide shows. Create a screen — a snapshot of your planned product as it runs. Anything goes: words, borders, box rules, inverse and underlining of monochrome, fore- and background color. Copy this "slide" to an empty screen. Change it a little, to show the next instant of run-time. Do it again. Presto, a whole slide show of your program in action.

All 250 characters and attributes are available from scrollable lists which pop to the screen. All commands are layered in Lotus-style pop-up menus. Frequent choices mapped to function keys as well. 80x25 character mode, not bit-mapped.

Screen areas can be blocked for cut and paste or filled with color or characters, even blink. Slides can overlay on others, can be shuffled, deleted. Slides can proceed at time intervals or branch anywhere in the slide sequence depending on user keyhits.

Invaluable to prototype the program you are about to write, to position the labels, choose the color decor, smoothe out the keystroke interface. Or load the "capture" utility and snapshot the screens of any running program for an instant slide show.

Each copy entitles you to redistribute fifty of the slide projector program that runs demos. Plain manual, no binder keeps price of big product small. "Might... become *the* essential tool in... user interface prototyping," *Tech Journal.* Ask for: N0100. List: **$75** US: **$69**

BASTOC OPTIMIZES!

Translates BASIC Into C

For a trifling price, BASTOC™ moves truckloads of BASIC code over to C. It's a translator which takes in Microsoft Extended BASIC and emits pure K&R C for Lattice 3.0. It will optionally convert your program into a single monolithic C function or decompose it into separate functions, one for each GOSUB label.

Version 2's optimization dramatically reduces execution time. Converts to integers those variables in BASIC programs which do not need floating point. Where BASIC uses full assignment statements to increment counters, BASTOC converts to C's compact form. Strings dynamically allocated ridding your application of BASIC's catatonic halts for garbage collection. Creates structure of even convoluted BASIC code. Huge worksaver.

Ask for:	List:	PC Brand:
S0375	**$495**	**$399**

Shopping List for the Power Workbench

	LIST	OURS
ASSEMBLERS & DEBUGGERS		
Advanced Trace-86 *Morgan, ASM Interpreter*	175	149
Codesmith-86 *Debugger by Visual Age*	145	109
Cdebugger *by Micro-Software Developers*	165	139
CSD Debugger *C source level by Mark Williams*	75	75
C-Sprite Debugger *by Lattice, source level*	175	139
Microsoft Macro Assembler *with Utilities*	150	109
Periscope I Debugger *Data Base Decisions*	295	269
Periscope II *Data Base Decisions*	129	111
Pfix86 *by Phoenix, Assembly level debugger*	195	149
Pfix86 Plus *by Phoenix, Symbolic Debugger*	395	279
BASIC LANGUAGE		
BetterBASIC *Summit Software*	195	165
BetterBASIC Utilities *8087 Math Support*	99	85
Btrieve Interface	99	85
Run-Time Module	250	225
Microsoft BASIC Interpreter *for XENIX*	350	295
Microsoft QuickBASIC Compiler *full BASICA*	99	79
Professional BASIC *by Morgan*	99	79
8087 Math Support	50	47
True BASIC *True BASIC Inc*	150	119
Run Time Module	500	420
True BASIC Libraries *Btrieve, Asyn, Sort, etc.*	Var	Call
C COMPILERS		
C-86 Compiler *Computer Innovations*	395	289
Lattice C Compiler *from Lattice*	500	299
Let's C Compiler *by Mark Williams*	75	69
with CSD Source Level Debugger	150	129
MWC-86: Mark Williams C Development	495	369
Microsoft C Compiler *4.0*	450	295
C INTERPRETERS		
C-Terp *by Gimpel Software*	300	249
Instant C *by Rational Systems*	500	395
Interactive-C *by IMPACC with debugging*	249	219
RUN/C Professional *from Lifeboat*	250	185
RUN/C *without Loadable Libraries*	120	109
TEXT EDITORS		
Brief *from Solution Systems*	195	CALL
CVUE *by Lattice*	75	69
with Source	250	195
Epsilon *by Lugaru Software, like EMACS*	195	169
Fast C *by Lifeboat*	130	105
FirsTime *by Spruce Technology, C syntax*	295	229
Kedit *by Mansfield, similar to Xedit*	125	115
LSE, the Lattice Screen Editor *Multi Window*	125	100
Pmate *by Phoenix, with Macros*	225	159
Text Management Utilities *Grep, splat, diff, etc.*	120	100
Vedit *by Compuview*	150	119
Vedit Plus *by Compuview*	225	180
FILE MANAGERS		
Btrieve *by Softcraft, no royalties*	250	195
Btrieve Network *by Softcraft*	595	465
c-tree *by FairCom — no royalties, source*	395	329
dBC *dBASE file manager from Lattice*	250	195
with source	500	390
dbVista *single user DBMS by Raima*	195	159
with source	495	429
dbVista *multi-user DBMS*	495	429
with source	990	849
Opt-Tech Sort *Can sort Btrieve files*	149	119
SCREEN DESIGN		
Curses *by Lattice, UNIX screen designer*	125	99
with Source	250	199
On-Line Help *from Opt-Tech Data*	149	119
Panel *by Roundhill, no royalties*	295	229
View Manager for C *by Blaise*	275	209
Windows for C *Vermont Creative Software*	195	149
Windows for Data *includes Windows for C*	295	250

	LIST	OURS
ZView *Data Management Consultants*	245	199
GRAPHICS		
Essential Graphics *by Essential, no royalties*	250	210
GSS Graphics Development Toolkit	495	399
GSS Kernel System *by Graphic Software*	495	399
GSS Kernel System *for IBM RT*	795	676
GSS Metafile Interpreter	295	249
GSS Plotting System	495	399
Halo Graphics Kernel System	300	219
with Dr. Halo II, by Media Cybernetics	440	299
COMMUNICATIONS		
Asynch Manager *by Blaise, for C or Pascal*	175	149
Greenleaf Communications *by Greenleaf*	185	139
Software Horizons Pack 3	149	119
UTILITY LIBRARIES		
Blaise C Tools Plus C Tools 2	175	149
Blaise C Tools	125	109
Blaise C Tools 2	100	89
C Food Smorgasbord *by Lattice*	150	109
C Utility Library *by Essential, 300 functions*	185	139
Greenleaf Functions *by Greenleaf Software*	185	139
PforCe *by Phoenix, vast library*	475	349
Software Horizons *Packages*	Var	Call
TopView Tool Basket *by Lattice, source avail*	250	199
Vitamin C *by Creative Programming*	150	139
DEVELOPMENT TOOLS		
Code Sifter *by David Smith Software, Profiler*	119	99
C-Worthy *by Custom Design Software*	295	269
C-Worthy for Network *Menus, help, errors*	495	449
Dan Bricklin's Demo Program *Prototyper*	75	69
LMK from Lattice *by Lattice, "make" like UNIX*	195	149
PC-Lint *by Gimpel Software, after UNIX's "lint"*	139	125
PFinish *by Phoenix, EXE performance analyzer*	395	279
Plink86 *by Phoenix, Overlay Linker*	395	279
Plink86 Plus *Utilizes memory for overlays*	495	359
Pmaker *by Phoenix, like UNIX "make"*	195	149
Pre-C *by Phoenix, UNIX "lint"-alike*	395	279
Pfantasy Pac *six Phoenix products*	1295	895
OTHER TOOLS		
BASTOC *by JMI, convert BASIC to C*	495	399
BASIC-C *BASIC's functions added to C*	175	139
The HAMMER *by OES Systems*	195	179
PASM86 *by Phoenix, Macro Assembler*	295	219
PTel *by Phoenix, Binary File Communicator*	195	149
Rtrieve *by Softcraft, Btrieve Report Generator*	85	75
Xtrieve *by Softcraft, Query Utility for Btrieve*	195	175
FORTRAN COMPILERS & UTILITIES		
ACS Time Series *by Alpha Computer Service*	495	469
Forlib- Plus *by Alpha Computer Service*	70	59
Microsoft FORTRAN *Links with Microsoft C*	350	219
Microsoft FORTRAN *for XENIX*	495	389
Pro FORTRAN *by Prospero*	390	345
RM/FORTRAN *by Ryan-McFarland*	595	399
Scientific Subroutine Library *by Peerless*	175	149
Scientific Subroutine Package *by Alpha*	295	269
The Statistician *by Alpha Computer*	295	269
Strings & Things *by Alpha Computer*	70	59
OTHER LANGUAGES & UTILITIES		
Microsoft COBOL Compiler	700	499
Microsoft COBOL Compiler *for XENIX*	995	795
Microsoft COBOL Tools *with Source Debugger*	350	259
Microsoft COBOL Tools *for XENIX*	450	333
Microsoft Lisp *New Common Lisp*	250	189
Microsoft MuMath *Includes MuSimp*	300	199
Microsoft Pascal Compiler *Links with M'soft C*	300	199
Microsoft Pascal Compiler *for XENIX*	495	415
Pro Pascal *by Prospero, ISO Validated*	390	345

PRICED TO SAVE YOU MONEY, SHIPPED FAST ANYWHERE.

BEST PRICES YET!

INTERACTIVE–C

NEW!

Compiler-Compatible Interpreter, Editor, Debugger

Earlier C interpreters were miraculous compromises. Interactive-C™ shows how far C interpreters have come. More than an interpreter, Interactive-C is a fully-integrated development environment: a complete K&R interpreter bound tightly to its own editor and debugger.

Slice through programming projects like a hot knife through butter. Extensive error-checking insures immediate detection of program misbehavior. State of the art debugging tools include breakpoints, watchvalues, several stepping options and interactive viewing and modification of variables. An Interactive-C exclusive lets you interrupt to edit and "continue" from where you left off. Eliminates plodding replays of already debugged code — the ball and chain of other interpreters.

Operate Interactive-C using adjustable edit, command, and status windows. Toggle a second screen showing only your program's output — never any crowded intermixing. Or, boost productivity with twin CRTs. Load object code of functions you have already compiled. Or of commercial libraries. Interactive-C has immediate mode, syntax checking both as you type and run, and cursor positioning precisely pointing at an error, not possible with incremental or pseudo-compilers which leave source code behind.

100% compiler compatible — right down to header files and library calls. Port programs between Interactive-C and your compiler with no modifications whatever — not even tricky areas of dynamic memory allocation and I/O.

Specify:	List:	PC Brand:
E950 & Compiler	$249	$219

RUN/C PROFESSIONAL

C Interpreter Links Binary Libraries

Run/C comes in an apprentice and pro version. The professional model dynamically loads and unloads multiple binary function libraries like C-Food Smorgasbord™ and Halo Graphics™ — potentially any library compiled with Lattice's large model. Inside this interpreter your C program can reach for functions in the best of commercial libraries.

This C interpreter behaves like PC BASIC meets WordStar®. Use fullscreen editing to create a program. RUN it. If it stumbles, LIST it, EDIT it, RUN it again, fix it again. Use familiar commands like LOAD, MERGE, SAVE, FILES, even TRON and TRACE.

Ideal for program development. Put up code at high speed, try out things devil-may care, let RUN/C find your malaprops. Blast away until tight little code segments are undyingly faithful.

Lots more features: system interrupts, a shell command to invoke any operating system command without leaving Run/C, debugging aids ingeniously installed as a Run/C function. Call for debugging conditionally from within your program, a specific function or a menu of aids including immediate mode, single-step tracing, changing of variable values.

Manual shows how to develop the interface to a commercial library, using the Lattice compiler (a must!). Link your own function archive the same way. (320k minimum; 512k recommended to fit libraries.)

Ask for: S0950 List: $250 PCB: $185

PLINK86 & PLUS

Cached Overlays Maximize Memory Use

Long the overlord of overlay linkers, standard Plink86 shoehorns large programs into small machines by sharing memory, swapping program segments in from disk. A 512k program could run in a 128k machine, for example. The Plus version goes beyond: if it finds itself in a larger machine, it moves whatever program overlays that fit into leftover memory. Overlays now swap at memory speed not disk speed. It also can automatically restore a displaced overlay to which a subsequently called overlay must return, and assign library modules to a program's root segment or its overlay areas.

Plink86-Plus: List: $495. Us: $359. Plink86: List: $395. Us: $279

LATTICE C COMPILER

Major Upgrades to the Best Selling C Compiler

Lattice now embraces key UNIX™ enhancements which have entered the language since K&R: void functions returning no value, enumerated data types to assign stepped values to variables, data passing between structures by assignment.

The greatly expanded libraries (325 functions!) enable the file sharing and record locking provisions of DOS 3.1, provide a full complement of transcendentals, and a host of utilities to mimic the UNIX and XENIX™ environments.

Lattice 3.0 defaults to the ANSI proposed standard when you need strict adherence, but command line options restore leniency. And it adopts ANSI checking of external function arguments by data type to kill bug swarms when modules join up at link time.

Lattice now delivers smaller .EXE files, boasts very fast link times and a more efficient aliasing algorithm. New options generate code to use 80186 and 80286 features; 8087 of course sensed and utilized. Lattice has enjoyed pre-eminence so long that developers have created far more snap-on tools for Lattice C than any other compiler. William Hunt's *PC Tech Journal* review of 12 compilers awarded Lattice the only "very good" rating for add-on library availability.

Ask for:	List:	PC Brand:
S0100	$500	$299

FOREIGN POLICY

We ship anywhere. Phone or Telex your order. Credit cards: We need card number, expiry date, name and address of card. Or wire funds to PC BRAND, c/o Chemical Bank, 126 East 86th St., NY NY 10028, Account No: 034-016058. **We'll ship immediately** and confirm by Telex if you provide number.

BETTER BASIC

Convert Microsoft BASIC. Structured, Compilable.

Combines the familiarity of BASIC with the best features of C, Pascal, and Modula 2, yet BetterBASIC is 100% compatible with Microsoft's GW™ BASIC and IBM BASICA including graphics, sound, and assembly language calls. So load your old programs and RUN. SAVE and they are converted automatically to BetterBASIC!

It's big: Needs 192k; programs can go to the PC's full 640k. It's comfy: Behaves like M'soft BASIC at the interactive level, with a full-screen editor, direct statement execution, and always poised to RUN. It's fast: Each statement checked and compiled once, not every time encountered. Sieve runs *6 times faster* than with M'soft.

C-like structures house file records so goodbye to FIELD, MKI$, CVD, LSET, etc. Named "procedures" replace GOSUBs to linenumbers. Lots more features: built-in linker for compiled modules; trace; debugging breakpoints; cross-reference command; 32k strings; DOS and BIOS calls and interrupts; recursion. Run-time module stores object code for redistribution.

Ask for::		List:	Us:
S1200	BetterBASIC	$195	$165
S1201	Run-time Module	$250	$225
S1202	8087 Interface	$ 99	$ 85
S1205	Btrieve Interface	$ 99	$ 85

GSS GRAPHICS SYSTEM

ANSI CGI STANDARD!

Leave the Device Driving to GSS

GSS™ has reconfigured two components of its comprehensive graphics tools to conform with the ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers house management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means your programming remains generic; just switch drivers and the same program will drive a different device.

GSS Kernel™ conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc. So powerful, a single command may represent several score lower level statements.

Plotting has the equivalent GKS tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel and Plotting have tools to convert images they create to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for recreation on various devices.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Needs 256k.

Ask for:		List:	PC Brand:
GS010	CGI Dvlpmt Toolkit	$495	$399
GS020	Kernel System	$495	$399
GS025	Kernel for IBM RT	$795	$676
GS030	Plotting System	$495	$399
GS040	Metafile Interpreter	$295	$249

BTRIEVE

ASK ABOUT XTRIEVE & RTRIEVE

Queen B-tree File Manager Abdicates Royalties

There's no longer a tithe to incorporate Btrieve™ in applications, a welcome proclamation if royalties would ruin your profit margins. Btrieve takes complete charge of all file creation, indexing, reading, writing, insertion, deletion; space recapture, forward and backward searching. It builds function call "commands" right into the language you use: interfaces to C, Pascal, BASIC, and COBOL, with sample programs in all four, come with each copy.

Btrieve has mainframe specifications! Its balanced-tree indexing scheme finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; indexes up to 255 characters; files of 4 billion bytes. Can even extend a file across two drives — even two hard disks!

Version 4.x speeds DOS interaction for large multiply-keyed files; enables variable length records of virtually any length; verifies accuracy (optionally) with read after write, useful in gritty environments; offers password and data encryption.

There's also Xtrieve, for Btrieve file inquiry and data manipulation, and Rtrieve for report writing. All three in versions for any network that supports the MS-DOS 3.1 file sharing function.

Ask for:	List:	PC Brand:
S0650	$250	$195
S0652 Network Version	$595	$465

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute its ownership. We will inquire for you about site licenses. Except as otherwise indicated or where "#" follows the Product Code, products may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM® PC, XT or AT under PC-DOS and require no more than 128k of RAM unless indicated. Non IBM machines using MS-DOS: contact manufacturer about precise differences so we can advise.

Returns: See box page one. Defective parts will be replaced. Please call for authorization to return a product for refund.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders, Literature, or Catalogs, Call Us at…

800 PC-BRAND

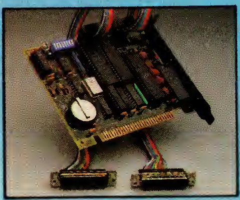
That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

Prices, terms, and specifications subject to change without notice.

SAVE $5
Just tell us this code:
J8

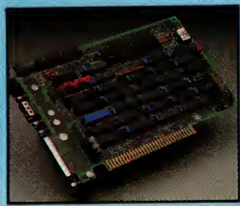
Easy Parking

When you're looking for efficient I/O expansion, it pays to think small.



QuadPort XT

Plug QuadPort XT into the short slot of your IBM XT or XT compatible and increase your port options with a printer port and a communications port. It's easy and efficient. QuadPort lets you attach a printer to your system, or any parallel device. Like a plotter or buffer. You can also add a modem for communications, or another serial device. Like a serial printer. Plus, QuadPort XT comes with a battery powered clock/calendar that can be set up to automatically feed your system clock whenever you boot up. An optional second serial port is also available.



QuadPort AT

For AT systems, there's QuadPort AT. QuadPort AT gives you a parallel printer port and a serial communications port just like the QuadPort XT. And features the same efficient short-slot design. But QuadPort AT goes one step further with built-in expandability. An optional daughter card snaps onto QuadPort AT to add another four serial ports to your AT system. Use these ports to expand with extra peripherals. Or add terminals when your AT becomes the center of multi-user system.

For more information, visit your nearest Quadram dealer. Or contact us at One Quad Way, Norcross, Georgia 30093; 404-564-5566.

CIRCLE 171 ON READER SERVICE CARD

QUADRAM®
An Intelligent Systems Company

Accelerated Keystrokes

The repeat-key action of the PC keyboard is slower than it should be but can be speeded up with software.

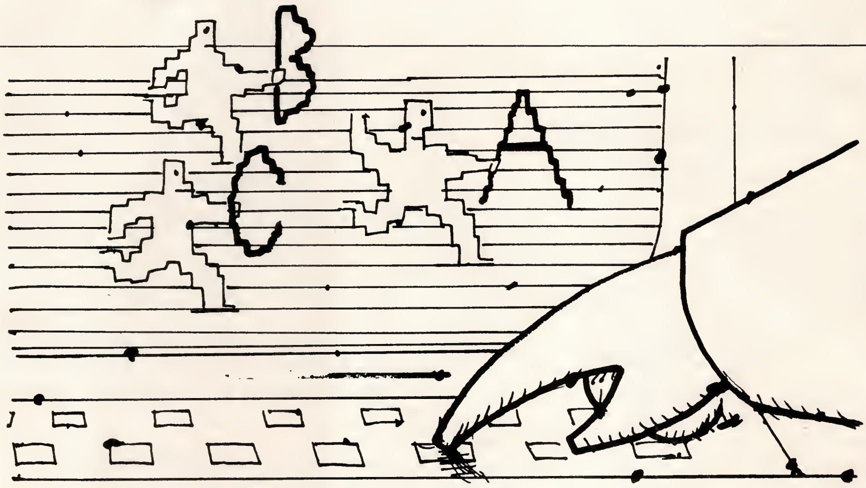


ILLUSTRATION • MACIEK ALBRECHT

The PC keyboard typically takes one-half second to begin repeating a keystroke. Once the autorepeat process has begun, it repeats keystrokes at the rate of 11 per second, taking about 7 seconds to move the cursor across the screen. This typematic action, which can cost the user valuable time, is not as quick as it could be.

The assembly language program that is shown in listing 1 (called KWIKKEY.ASM) uses the computer's timer and interrupt mechanisms to increase the keyboard's repeat-key rate. This, in turn, decreases the time the user spends waiting for the keyboard to respond to keystrokes. The program can be used with a PC, PC/XT, or PC/AT.

KEYBOARD BASICS

KWIKKEY.ASM is able to accelerate the PC keyboard because of the flexible design of the PC's keystroke handling. The keyboard contains its own Intel 8048 microprocessor. A control program in the ROM of the 8048 controls the processor's actions. This program is responsible for identifying individual keystrokes. It maintains communication with the computer as it serializes data for output to the system.

When a key is pressed (called a *make*), the keyboard sends the computer a scan code, ranging from 1 to 83; each represents a different key on the keyboard. When a key is released (called a *break*), the keyboard adds 128 (80H) to the original scan code and sends the resulting code to the computer. For example, the scan code for the Esc key is 1. When Esc is pressed, the keyboard sends a 1 (01H) to the computer; when Esc is released, the keyboard sends a 129 (81H).

When the keyboard senses either a make or a break, it signals the 8259 interrupt controller chip, which is located at the PC end of the coiled keyboard cable, to interrupt the action of the CPU. The 8259 then forces an INT 9, which passes control to the KB_INT procedure contained in the ROM BIOS. This routine reads the scan code from the keyboard and responds appropriately.

The KB_INT procedure is very sophisticated. It must handle the cases of Shift, Ctrl, and Alt key combinations, as well as the special commands Shift-PrtSc, Ctrl-Break, Ctrl-NumLock, and Ctrl-Alt-Del. As soon as the keystroke has been interpreted, the KB_INT procedure places an ASCII (or special extended ASCII) value in a circular queue. When a program requests keyboard input, it is passed the first keystroke contained in this buffer.

Because of this flexible system, nearly every facet of the PC's keyboard handler can be modified. The most common way to modify keyboard action is to replace the BIOS KB_INT routine with a custom driver. Several such keyboard utilities are available, such as SuperKey from Borland International and ProKey from RoseSoft.

However, simply replacing the BIOS KB_INT routine does not affect the typematic action of the PC keyboard. KB_INT only processes each scan code as it comes from the keyboard. The 8048 processor in the keyboard is responsible for determining how soon to start the repetitions as well as how fast to send them.

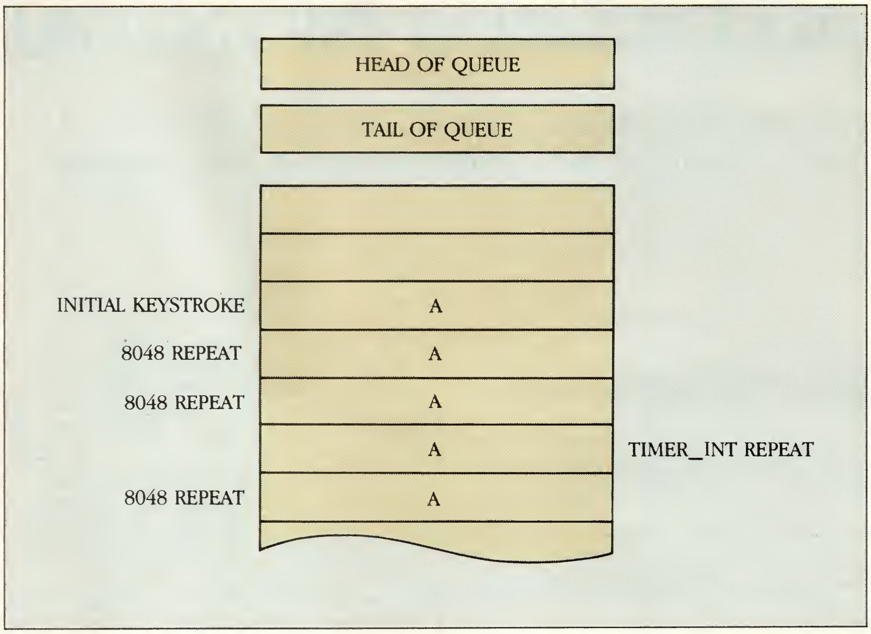
TIMER TICK TRICKS

In much the same way that the keyboard interrupts the CPU when a scan code is ready, the realtime clock also forces hardware interrupts. It does not wait for an external event, such as a keystroke, but interrupts the CPU on a regular cycle, every 55 milliseconds or about 18.2 times a second. The interrupt vector called the timer tick (INT 1CH) can be used to intercept the realtime clock interrupt. Spooler programs and other background tasks often tie into the timer tick in order to do a slice of processing in the background while another program continues running.

KWIKKEY.ASM also uses the timer tick interrupt. The program intercepts both the keyboard interrupt (KBD_INT) and the timer interrupt (TIMER_INT) and makes them work together to handle repeat-key actions.

The version of KBD_INT defined in KWIKKEY.ASM monitors the keyboard buffer. When it senses that a keystroke has been recorded, it saves the keystroke in a memory variable. Simultaneously, the TIMER_INT included in KWIKKEY checks continuously to see if a repeat of the keystroke is desired. At the interval specified by the user at the beginning of the program, TIMER_INT stores the saved keystroke in the keyboard buffer, effectively repeating the

FIGURE 1: *Keyboard Buffer*



When a key is held down, the 8048 keyboard controller adds keystrokes to the BIOS keyboard buffer. The KWIKKEY.ASM program (see listing 1) uses the timer interrupt to add 20 key repeats every second.

previous key—long before the keyboard itself would have been able to repeat the key. The net result is that keys can be repeated at up to three times the normal rate.

The KWIKKEY.ASM program is written as three separate procedures: SET_UP, KBD_INT, and TIMER_INT. When KWIKKEY is executed, it jumps over the data and the interrupt handlers to get to the SET_UP procedure, which is the initialization portion of the program. SET_UP first checks to make sure that a copy of SET_UP has not been installed already.

Any program that intercepts an interrupt risks the possibility of interfering with other programs that perform the same task. To avoid this danger, KWIKKEY does not disable the system interrupt handlers. It simply borrows the interrupts temporarily, always passing control down the line to the original interrupt handler. If another timer interrupt handler has been installed, this technique makes sure that the original handler is given its expected time share. If KWIKKEY proves to be incompatible with some already installed program, the answer might be to install KWIKKEY first.

After storing the original keyboard and timer interrupt vectors in its data area, KWIKKEY loads the new KBD_INT and TIMER_INT addresses. The program then uses the INT 27H "terminate-but-stay-resident" service to pass control back to DOS, leaving the interrupt handler code resident in memory.

After both interrupt handlers have been installed, TIMER_INT is executed 18.2 times per second. Thus, its action can be considered continuous; it is executed regardless of any other processes taking place. KBD_INT, on the other hand, is executed only when the user presses or releases a key.

When a key is pressed or released, KBD_INT checks the current position of the end of the queue in the keyboard buffer (called the *buffer tail*). (See figure 1 for a diagram of the keyboard buffer.) After checking the buffer tail, KBD_INT then invokes the original keyboard interrupt. Upon returning, it checks the position of the buffer tail again. If the position has changed, it knows that a keystroke has been input in the interval. The new keystroke is saved for handling later by the timer interrupt, and the variable INIT_DELAY is initialized to start the countdown before the first repeat. Note that the repeat mechanism is disabled in the special case of an artificial keystroke, such as Alt combined with a key from the numeric keypad.

However, if the buffer tail has not moved since KBD_INT first checked its position, the interrupt must have been caused by a Shift key combination (which does not produce an ASCII

code) or by the release of a key. In either case, a keystroke repeat is not necessary, and a flag is set to prevent the timer interrupt from taking any action. Any autorepeat in progress is stopped. The interrupted program then is exited via an IRET instruction.

At the same time, the TIMER_INT continues to be executed 18.2 times per second. Its job is to simulate keyboard activity by placing scan codes in the keyboard buffer at appropriate times. It checks the value of the INIT_DELAY variable continuously. If the value is not 0, it decrements it and exits without taking any further action. The effect is that a new keystroke is not repeated until the fifth time that TIMER_INT is executed. Because the timer ticks at a rate of about 55 milliseconds, the keystroke does not begin repeating for about 275 milliseconds (or about one-quarter of a second). A longer delay is not necessary; this is plenty of time to allow the user to release the key. A shorter delay might cause spurious repeats. Users can experiment with this value by changing the REPT_DELAY equate, which is located at the beginning of the KWIKKEY.ASM program.

After the one-quarter second delay, TIMER_INT begins counting down to the time specified at the beginning of KWIKKEY.ASM to put a repeat keystroke in the keyboard buffer. The RATE_DELAY variable is decremented with each subsequent pass. When it reaches 0, the scan code in LAST_KEY is placed in the keyboard buffer. If the buffer is full, no action is taken.

In the program shown in listing 1, the REPT_RATE constant is set at 1, which means that a repeat is forced once every two executions of the TIMER_INT. The user can change this constant to force the repeat to occur at a different interval. TIMER_INT is executed 18 times per second; as a result, 9 repeats are added to the 11 that come from the keyboard, for a total of 20 repeats per second. This rate can be changed with the REPT_RATE equate at the beginning of the program.

WORKING WITH AN AT

While the KWIKKEY.ASM program can be used with an AT, a much simpler method also exists for accelerating the reaction time of the AT's keyboard. This is because the AT serial keyboard link is bidirectional. This allows data to be sent directly to the 8048 keyboard controller. As a result, repeat-key action can be augmented with a simple BASIC program called KWIKAT.BAS (see listing 2) or a short assembly language program (see "Rev Up the AT Keyboard," Tech Notebook 37, Kevin M. Crenshaw, May 1985, p. 39).

The KWIKAT.BAS program can be included in an AUTOEXEC.BAT file. KWIKKEY.ASM proves the value of assembly language; it could not have been written in any other language. The program also illustrates how to install a timer-driven interrupt handler using a technique that does not interfere with other programs of the same nature.

KWIKKEY demonstrates the concepts of background tasking and coroutines. While the TIMER_INT and the KBD_INT procedures are useless when examined individually, the dynamic interplay between the routines provides a unique software solution to a shortcoming of the PC keyboard.

Dan Rollins is a computer consultant and freelance technical writer. He is the author of the book, IBM PC: 8088 Macro Assembler Programming *(Macmillan, 1985).*

LISTING 1: KWIKKEY.ASM

```
;----------------------------------------------------------------
; KWIKKEY.ASM  Version 1.4                              5-28-86
; Copyright (c) 1985 by Dan Rollins
;
; This program speeds up the repeat action of the IBM PC and XT
; keyboard. After installation, a keystroke begins repeating about
; 1/4 second after the initial keystroke and the repeats occur at
; about twice the normal speed. These delay and rate parameters may
; be modified. The program uses the 55ms timer interrupt to augment
; the speed of the keyboard. The basic idea is to have the timer
; interrupt check to see if the key has been released. If not, then
; it stuffs a keystroke into the BIOS keyboard buffer. Notes: This
; program must be used with DOS 2.x or later.  It is a COM format
; program, so it must be processed by EXE2BIN.
;----------------------------------------------------------------
;=== program equates ===

REPT_DELAY    equu    5  ;number of 55ms intervals to skip before
                         ; the first repeat.  5 = 275ms = about
                         ; 1/4 second.   Use at least 2 to avoid
                         ; "key bounce"

REPT_RATE     equ     1  ;Select from:   0 = 29 repeats per second
                         ;               1 = 20 repeats per second
                         ;               2 = 16 repeats per second
                         ;               3 = 13 repeats per second
                         ;       4 or more = standard repeat rate
BIOS_DATA_SEG equ     40h  ;These addresses are listed
BUF_START     equ     1eh  ; in the Technical Reference manual
BUF_END       equ     3eh
BUF_HEAD_ADDR equ     1ah
BUF_TAIL_ADDR equ     1ch
ALT_NUM_BUF   equ     19h

FALSE         equ     0
TRUE          equ     1

;========================
com_seg   segment
          assume  cs:com_seg, ds:com_seg
          org     100h                    ;must set up for COM file
kwikkey   proc    far
          jmp     set_up      ;get past the data and install the
                              ;interrupt handlers

;============= program data area ========
delay          db     REPT_DELAY  ;max ticks BEFORE STARTING to repeat
rate           db     REPT_RATE   ;maximum ticks BETWEEN repeats

inst_flag      dw     1234h    ;KWIKKEY signature when already installed
rep_ok         db     FALSE    ;flag turns off repeat while processing
last_key       dw     0        ;stores most recent keyboard scan code

init_delay     db     REPT_DELAY  ;remaining ticks before
                                              ;starting to repeat
rate_delay     db     REPT_RATE   ;remaining tick between repeats

bios_kbd_int   label dword    ;DWORD so it can be used in a FAR call
bki_offset     dw     0       ; This is set to the addr of BIOS KB_INT
bki_segment    dw     0       ; at the time of installation of KWIKKEY

user_timer_int label  dword   ;used to preserve "forward chain" of user
uti_offset     dw     0       ;timer interrupt handlers.  Set in SET_UP
uti_segment    dw     0       ;procedure, this addr will normally point
                              ;to an IRET.


;---------------------------------------------------------------------
; KBD_INT
; This procedure intercepts keystrokes and sends control to normal
; BIOS KB_INT.  Its primary function is to set up for the repeat action
; that occurs in the TIMER_INT.  It checks each key that comes in, and
; resets a delay counter if it is a new keystroke.
kbd_int  proc    far
         mov     cs:rep_ok,FALSE         ;turn off repeats while
                                         ;processing
         push    ax                      ;save the registers
```

```
        push    si
        push    ds
        mov     ax,BIOS_DATA_SEG      ; set up to address
        mov     ds,ax                 ; BIOS data area
        mov     si,ds:[BUF_TAIL_ADDR] ;get addr of current
                                      ;buffer tail
       ;check special case: don't repeat ALT-numpad keystrokes

        cmp     byte ptr ds:[ALT_NUM_BUF],0 ;is one in progress?
        je      ki_10                       ; no, continue
        pushf                               ; yes, process
        call    cs:bios_kbd_int             ; the keystroke
        jmp     ki_exit                     ; exit with repeats off
ki_10:
        pushf                         ;simulate a normal interrupt
        call    cs:bios_kbd_int       ; and process the keystroke

        cmp     si,ds:[BUF_TAIL_ADDR] ;did the tail move?
        jne     ki_20                 ; no, either a shift-key
                                      ; or a release
        mov     cs:last_key,0         ; insure no spurious repeats
        jmp     ki_exit               ; exit with repeats turned off
ki_20:                                ; yes, process the keystroke
        mov     ax,[si]               ; get new scan code
        cmp     ax,cs:[last_key]      ; same as last time?
        mov     cs:last_key,ax        ; (save the key for next time)
        je      ki_30                 ;  yes, hardware repeat
                                      ;       just reset the rate
                                      ;  no, reset both timer delays
        mov     al,cs:delay           ;  get maximum tick count
        mov     cs:init_delay,al      ;  set delay before repeat
ki_30:
        mov     al,cs:rate            ;    get maximum tick value
        mov     cs:rate_delay,al      ;    set delay between repeats
        mov     cs:rep_ok,TRUE        ;  OK to continue repeating

ki_exit:
        pop     ds
        pop     si
        pop     ax
        iret                          ;also restores Flags of interruptee
kbd_int endp

;---------------------------------------------------------------
; TIMER_INT
; This procedure is executed 18.2 times per second
;
; It checks to see if a repeat is needed.  If so, it stuffs the scan
; code into the keyboard buffer, ready for the next keystroke request
;
timer_int proc far
        cmp     cs:[rep_ok],TRUE    ;are repeats blocked?
        jne     ti_exit             ;   yes, resume without repeat

        cmp     cs:init_delay,0     ;finished delaying before
                                    ;first repeat?
        je      ti_10               ;   yes, check rate delay
        dec     cs:init_delay       ;   no, decrement timer
        jmp     ti_exit             ;       and resume without repeat

ti_10:
        cmp     cs:rate_delay,0     ;finished delaying between repeats?
        je      ti_20               ;   yes, do the repeat
        dec     cs:rate_delay       ;   no, decrement rate timer
        jmp     ti_exit             ;       and resume without repeat

;------ repeat the previous keystroke ----------

ti_20:
        push    ax                  ;save all registers used
        push    si
        push    ds
        mov     ax,BIOS_DATA_SEG  ;prepare to address BIOS data area
        mov     ds,ax
        mov     ax,ds:[BUF_TAIL_ADDR]  ;get current position
                                       ;in kbd buffer
        mov     si,ax                  ;we'll need this address later
        add     ax,2                   ;point to next position in buffer
```

With one Hayes Transet 1000® you can do three things at once.

Think

You can keep working with your computer.

Communicate

While receiving electronic messages from your modem.

Print

While your printer is printing another job.

*Manufacturer's estimated retail price.

We call it triple tasking.™ Transet 1000 helps your productivity, by allowing you to perform three separate computer tasks. Simultaneously. No more waiting for one task to be completed before you can begin another. And no more wasted time!

Transet 1000 keeps working for you—even when your computer is turned off. Because it has its own independent memory, Transet 1000 can serve as an electronic mailbox. Your files and incoming messages received through your modem remain stored overnight, or whenever you're away from the office. When you return, you can print out your mail without turning your PC on. Or, read it on your PC screen and print out only selected messages. You can even read your mail from any remote PC or terminal with a modem.

Before now, you would have had to buy several add-on devices to do all this. And *that* could be costly.

But no more. Transet 1000 does it all—for a surprisingly low price. It costs only $399* for the 128K model, which stores up to 90 pages. And only $549* for the 512K version, with up to 360 pages of storage.

So wait no more. If you need this flexibility in your operation, you should have a Transet 1000. See your authorized Hayes dealer for a demonstration. Or contact Hayes for information at 404-441-1617.

Hayes Microcomputer Products, Inc., P.O. Box 105203, Atlanta, Georgia 30348.

Hayes®

Say yes to the future

CIRCLE NO. 155 ON READER SERVICE CARD

Save 50%

TECH JOURNAL

Subscribe to the only IBM PC magazine written for systems designers and integrators and save 50%

Enter my subscription to PC TECH JOURNAL for:

☐ One year (13 issues) for only $26.70.

☐ Two years for only $53.35. **SAVE 50%!**

Savings based on annual single copy price of $53.35.

Mr./Mrs./Ms. ______________________________
please print name in full 4S523

Company ______________________________

Address ______________________________

City ______________________________

State ______________ Zip ______________

Add $6. per year for postage in Canada and all other foreign countries—Cash payment U.S. currency only. Basic annual subscription price is $34.97. Please allow up to 30 to 60 days for delivery of first issue. *Send no money—we will bill you later!*

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT #66 BOULDER, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

TECH JOURNAL
P.O. BOX 2966
BOULDER, COLORADO 80321

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT #66 BOULDER, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

TECH JOURNAL
P.O. BOX 2966
BOULDER, COLORADO 80321

```
        cmp     ax,BUF_END              ;past end of buffer?
        jne     ti_30                   ;  no, continue
        mov     ax,BUF_START            ;  yes, next position is the
                                        ;start
ti_30:
        cmp     ax,ds:[BUF_HEAD_ADDR]   ;if tail=head, buffer is full
        je      ti_40                   ;full, continue without repeat
        cli                             ;not full, dont allow break-in
        mov     ds:[BUF_TAIL_ADDR],ax   ;       update buffer position
        mov     ax,cs:[last_key]        ;       fetch key to repeat
        mov     [si],ax                 ;       store key in buffer
        sti                             ;       interrupts ok now
ti_40:
        mov     al,cs:rate              ;get the max rate delay value
        mov     cs:rate_delay,al        ;don't repeat for a while
        pop     ds                      ;restore registers
        pop     si                      ; and exit
        pop     ax
ti_exit:
        jmp     cs:[user_timer_int]     ;continue with previously
                                        ;installed interrupt handler
timer_int endp

LAST_BYTE equ   offset $+1  ;this is the address passed to INT 27H
                            ; Notice that the code of the SET_UP
                            ; procedure does not need to be preserved

;-----------------------------------------------------------------
; SET_UP
; This routine is executed only once -- when the program is installed.
;
; It resets the vectors of the KBD_INT and the USER_TIMER_INT,
; pointing them to code within this program.  Note that the original
; vectors are saved and executed so all previously-installed interrupt
; handlers remain operational.

logo_msg db 201, 25 dup(205),187,0dh,0ah
        db 186,'    KWIKKEY  Ver. 1.4    ',186,0dh,0ah
        db 186,' (c) 1986 by Dan Rollins ',186,0dh,0ah
        db 200, 25 dup(205),188,0dh,0ah,'$'

err_msg  db 'Error: ',07,'KWIKKEY already installed',0dh,0ah,'$'

set_up  proc    near
        ;------ first, make sure KWIKKEY hasn't been installed ---
        mov     al,9
        mov     ah,35h              ;DOS GET_VECTOR service
        int     21h                 ; for interrupt 9

        cmp     es:inst_flag,1234h  ;has KWIKKEY been installed?
        jne     su_10               ; no, continue
        mov     dx,offset err_msg   ; yes, display
        mov     ah,9                ;       error
        int     21h                 ;         message
        int     20h                 ;and exit to DOS
su_10:
        mov     al,9            ;get original vector of keyboard int 9
        mov     ah,35h          ;DOS GET_VECTOR service
        int     21h
        mov     bki_segment,es ;save original address
        mov     bki_offset,bx  ; so we can resume normally
                                              ; after intercept
        mov     dx,offset kbd_int
        mov     al,9            ;set vector for INT 9
        mov     ah,25h          ;DOS SET_VECTOR service
        int     21h

        mov     al,1ch          ;the user timer interrupt
        mov     ah,35h          ;DOS GET_VECTOR service
        int     21h
        mov     uti_segment,es ;save old address
        mov     uti_offset,bx  ; so we don't hog the timer interrupt

        mov     dx,offset timer_int
        mov     al,1ch         ;set vector to point to new TIMER_INT
        mov     ah,25h         ;DOS SET_VECTOR service
        int     21h

        ;------ display logo to indicate installation complete
        mov     dx,offset logo_msg
        mov     ah,9
        int     21h

        mov     cs:rep_ok,TRUE   ; it's OK to start repeat action

        ;------exit to DOS, leaving the interrupt handlers resident
        mov     dx,LAST_BYTE
        int     27h
set_up  endp
kwikkey endp
com_seg ends
        end     kwikkey         ;must specify for COM-format file
```

LISTING 2: KWIKAT.BAS

```
0 '*  AT keyboard repeat speedup sets the delay and repeat
1 '*  then prints the values according to formulas found
2 '*  in the AT Technical Reference
3 '*
4 '*  The programming sequence is as follows: first output a command
5 '*  (0F3h) then output a delay-and-rate value.  Bits 5 and 6 control
6 '*  the delay ranging from 250 ms up to 1 second.  Bits 0 through 4
7 '*  identify the repeat rate, ranging from about 30 repeats per
8 '*  second to 2 repeats per second.
9 '*
10 PRINT "  AT keyboard speedup"
15 PRINT "=low values are fastest="
20 INPUT "Initial delay (0-3): ",ID
30 INPUT "Repeat rate (0-31):  ",RR
40 OUT &H60,&HF3 :OUT &H60,(ID*32)OR RR
45 '** All set. Now calculate speed.
50 PRINT "initial delay is";
60 PRINT (ID+1) * .25;"seconds"
70 A=(RR AND 7) :B=(RR AND 24)\8
80 P=(8+A) * (2^B) * .00417
90 PRINT "repeat rate is";
95 PRINT 1/P;"per second"
```

PC Magazine Product of the Year

Macro Assembler

The quickest. Bar none.

Our Macro Assembler has long been the most complete package on the market. Now it's also the fastest. Three times faster than before. And faster than anyone else. Period.

Of course, it's still the most powerful assembler on the market. It supports the standard 8086/8087 opcodes. And the new 186/286/287 instruction set. So you can make the most of the new machines.

Debugging is quicker, too. Thanks to our interactive symbolic debugger, SYMDEB. Now you can refer to variables and source code instead of getting lost in hex dumps. And this debugger also works with Microsoft languages like C, FORTRAN and Pascal. So now you can set breakpoints and trace execution—using source code for reference.

Cut your development time dramatically. Microsoft Macro Assembler's Symbolic Debug utility lets you debug your Macro Assembler programs, or debug your Microsoft C, FORTRAN or Pascal programs using your original source code or the resulting disassembly. For example, you can set breakpoints on line numbers and observe the contents of variables or expressions.

SYMDEB is just part of our complete set of utilities. Tools that make programming as fast as it should be. There are the linker and library managers you'd expect. Plus a new version of MAKE, our maintenance utility, with improvements like macro expansions and inference rules.

We've also revised the manuals. Our new Macro Assembler has a lot to offer, so we added more examples. Now our manuals are not only thorough, they're clearer than ever before.

For quick development and assembly, the choice is obvious. Microsoft. There's nobody faster.

Microsoft® Macro Assembler Version 4.0 for MS-DOS®

Macro Assembler
- Fastest macro assembler for MS-DOS computers.
- Supports the 8086/8087/8088 and the 186/286/287.
- Define macros.
- Conditional assembly.
- Optional case sensitivity for symbols.
- 100% upward compatibility from earlier versions of both the Microsoft and IBM® Macro Assemblers.

Interactive Symbolic Debug Utility
- Source level debugger for programs written in Microsoft Macro Assembler, C Compiler, FORTRAN, and Pascal.
- Screen swapping helps debug highly visual applications.
- Set breakpoints on line numbers and symbols.
- Single step to follow program execution.
- Disassemble object code.
- Display and modify values.
- Full I/O redirection.

Program Maintenance Utility
- Rebuilds your applications after your source files have changed.
- Similar to UNIX™ MAKE utility.
- Supports macro definitions and inference rules.

Library Manager
- Create, organize and maintain your object module libraries created with Microsoft languages.
- Set page size from 16 to 32678, to create compact and granular libraries.

Object Code Linker
- Simple overlaying linker combines relocatable object modules created using Microsoft languages into a single program.
- Load Map generation.
- Specify from 1 to 1024 segments.

Cross-Reference Utility
- Creates a cross-reference listing of the definitions and locations of all symbols used in an assembly language program, which makes debugging programs easier.

Microsoft EXE File Compression Utility
- Packs EXE files for smaller size on disk and faster loading at execution time.

Microsoft EXE File Header Utility
- Display and modify EXE file header, allowing you to tune the stack size and initial memory allocation.

For the name of your nearest Microsoft dealer call (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Microsoft

The High Performance Software™

Reviews and Updates

MODULA-2 SDS
Interface Technologies Corporation

MODULA-2/86 BLS
Logitech Corporation

MIX C
MIX Software

MODULA-2 SOFTWARE DEVELOPMENT SYSTEM

Interface Technologies Corporation
3336 Richmond, Suite 200, Houston, TX 77098
713/523-8422

PRICE: $80.88

CIRCLE 341 ON READER SERVICE CARD

The Modula-2 Software Development System (M2SDS) from Interface Technologies Corporation (ITC) is a product with a troubled past. It has some redeeming qualities, however, that make it worthy of consideration.

As originally shipped (several months past the delivery date ITC had promised) it was incomplete, bug-ridden, and virtually unusable.

ITC remedied M2SDS's most serious omissions within a few months of shipping; however, the program remained bug-ridden for some time. Both versions 1.0 and 1.1 cause frequent system failures that necessitate power-off rebooting. These versions also are capable of corrupting library files on a disk and thereby destroying saved source code. System crashes are not uncommon even when using version 2.0cv (which is the most recent version, released in mid-April of 1986). For example, using ITC's Length(StringVariable) function as the upper bound of a FOR loop causes the compiler to freeze. In addition, compiling the following five lines of code causes the entire M2SDS environment to halt and display a CASE LABEL NOT FOUND error; unsaved work is destroyed, and the user is returned to the DOS command line:

```
MODULE LongBomb;
VAR
  L1: LONGINT;
BEGIN
  L1 := LONGINT(1 - L1);
END LongBomb.
```

Despite these pitfalls, however, M2SDS boasts some admirable characteristics, and a careful user can avoid the remaining bugs. For two reasons in particular, M2SDS deserves more attention than it has received.

First, M2SDS generates fast code. The timings in table 1 reveal that M2SDS and Logitech's Modula-2/86 generate code at approximately the same speed. Both generate code much faster than Turbo Pascal and much slower than Microsoft Pascal. (For a complete review of the Logitech compiler, see "Modula-2/86 Base Language System," Product Watch, this issue, p. 188.)

However, standard benchmarks do not give an accurate indication of how fast a large program actually will run. In comparing two nearly identical versions of a very large program (370KB of source code), our personal experience has shown ITC's M2SDS to be noticeably faster than the Logitech Modula-2/86 version. (Both are much faster than a smaller Turbo Pascal version of the same program.)

In addition to generating fast code, M2SDS also provides a very flexible memory model. Each module in an M2SDS program can have a 64KB code segment, a 64KB stack, and a 64KB data segment. Thus, these three segments of the program as a whole are limited only by available memory. All modules share a single heap (for dynamically allocated variables); however, the heap also can occupy all of available memory. M2SDS provides an overlay system that loads the code for an entire module onto the calling module's stack, then unloads it when the overlay module finishes executing. M2SDS produced native-code .EXE files that can occupy any amount of disk space (unlike Turbo Pascal's 64KB .COM files).

M2SDS provides a number of features not found in either Turbo Pascal or Logitech's Modula-2/86. M2SDS includes full source code for all its library modules, including a full DOS EXEC function and various low-level memory manipulation procedures. The product supports long (32-bit) integers. It also provides a Pascal-like STRING type that uses the 0 byte to represent the string's length. In all other respects, this type functions in a manner similar to that of Modula-2's standard null-terminated ARRAY OF CHAR type.

On the other hand, M2SDS still adheres to the severely restrictive set of CHARACTER/STRING compatibility rules presented in the second edition of Niklaus Wirth's *Programming in Modula-2* (Springer-Verlag, 1982). Thus, neither character variables nor single-character string constants can be passed to M2SDS procedures that have STRING or ARRAY OF CHAR parameters. The third edition of *Programming in Modula-2* specifies that characters be treated as single-character stirngs when passed to ARRAY OF CHAR parameters. Sensible implementations of the language should have made this change without waiting for Wirth to ratify it. (Logitech did.)

Because Logitech has reduced the price of its excellent Modula-2/86 from $500 to $89, M2SDS's place in the market is questionable. M2SDS does offer features that the Logitech compiler does not, but a comparative review would list more important advantages on the Logitech side of the equation.

M2SDS's superior user interface, however, may be enough to secure the

product's place in the market. Its system of windows and pull-down menus is well designed and easy to use. The product has a library system that stores everything connected with a module (including source and object versions of the definition and the implementation modules) in one place, which makes it easier to manage the great number of small files that Modula-2 compilers and linkers typically require.

More importantly, the M2SDS development environment uses a fully syntax-directed editor. It is quite inflexible—for example, it does not allow code to be moved to and from comments, and it does not allow changes in the basic structure of statements. As a result, a serious programmer could not rely on it exclusively. The editor does prevent syntax errors, however, and eliminates much of the tedium and frustration that poor typists can encounter when programming.

M2SDS's documentation is easily the best introduction to Modula-2 available. However, it says little about the technical details of ITC's implementation. Logitech's documentation says next to nothing about the language itself. It focuses instead on the technical details of the implementation. As the documentation indicates, M2SDS and Logitech's Modula-2/86 are aimed at very different markets, both legitemate.

For the serious programmer, M2SDS cannot compete with the Logitech compiler. Nonetheless, ITC's M2SDS should be appreciated for what it is: a powerful, inexpensive Modula-2 compiler with a good user interface. It is well suited to newcomers to Modula-2 and those programmers planning to work with the product only occassionally.

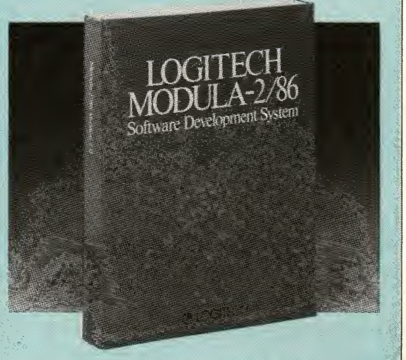
—COLE BRECHEEN
CHARLES BRADFORD

MODULA-2/86 BASE LANGUAGE SYSTEM

UPDATE

Logitech Corporation
805 Veterans Blvd., Building 201,
Redwood City, CA 94063
415/365-9852

PRICE: $89



CIRCLE 342 ON READER SERVICE CARD

Logitech has released version 2.0 of its Modula-2/86 Base Language System (BLS). The Modula-2/86 package, which consists of the MOD.EXE editor, three diskettes containing the runtime system, the compiler, the linker, and the libraries, and a manual, provides all the basic development tools necessary for a programmer to begin using Modula-2. (For a review of version 1.0 of this compiler, see "Modular Implementations," Tom Woteki, Alan Freiden, Dov Levy, Thor Bestul, and Robert Stine, December 1984, p. 154.)

The Logitech compiler and libraries are a complete implementation of Modula-2 for the IBM PC. Logitech has incorporated useful, though nonportable, syntax extensions, permitting abso-

TABLE 1: *Benchmark Timings*

TEST	ITC M2SDS 2.0CV	LOGITECH MODULA-2/86 2.0
LOOPS		
REPEAT	53	49
WHILE	48	51
FOR	41	35
CARDINAL ARITHMETIC INDEXING	148	150
1-dimensional array	164	167
2-dimensional array	121	134
PROCEDURE CALL		
0 parameters	106	102
4 parameters	181	183
BLOCK MOVE	216	216
POINTER CHAIN	268	200
15-BY-15 MATRIX MULTIPLY	8	5
FILE WRITE (30,000 characters to RAM disk)		
Character by character	145	19
Strings of 50	5	2
SCREEN WRITE		
10,000 characters	14	21
1,000 integers	9	9

All times in seconds.

The most striking performance difference lies in character-by-character file I/O: ITC's near-800 percent handicap reflects poor design in its runtime library.

lute address specification for data and pointer arithmetic.

MOD.EXE, the multifile, multi-window editor bundled with the BLS package, displays an elegant screen free of the clutter found in other editors' screens. A window status line labels the file in each window and notes if the file has been altered. The user interacts with the editor using pop-up menus and function keys. The cursor keys are similar to those in other editors, with some frustrating differences. Indentation of source code appears to be automatic, but MOD can be fooled by deeply nested block structures. A help screen can be customized by the user.

MOD is not a true syntax-assisted editor—it does not know the complete syntax of Modula-2 but can select the important elements to allow automatic indents. To minimize the number of keystrokes that must be typed by the user, a file configures MOD with syntactic skeletons that can be called up using Alt-key combinations. The user then fills in the blanks.

The compiler and linker can be called from MOD, but 512KB of available RAM is necessary for compiling large programs. A single keystroke commands MOD to position at compiler-detected errors. A syntax checker sensitive to indentation is invoked by pressing F2. However, indentation is not a component of Modula-2 syntax.

The compiler and linker in the BLS package are not in DOS.EXE format. They require DOS to load M2.EXE (the runtime system). M2.EXE in turn dynamically loads and executes LOD files, Logitech's version of .EXE files. M2.EXE also contains the interface between Modula-2 and DOS, which provides runtime support and error interception for Modula-2 programs. The command line that invokes M2.EXE allows the user to control the amount of memory allocated to the Modula-2 system.

Like some C compilers, this compiler passes through the source four times. During these four passes, it posts its progress on the console. Errors are reported on the console and in an error-listing file with a terse message describing the error and the offending source line. Console logging is optional after the first eight errors; unless they are suppressed, all errors are reported in the listing file. If the compiler was invoked through MOD, errors are passed back to MOD.

The compiler output depends on the type of file being compiled. LNK (Logitech object) files are generated for program and implementation modules. Definition modules yield SYM files, which contain symbolic information

about imported modules. The compiler also produces REF files, which are used by the optional debuggers.

Compiler options control listing generation, file search strategy, floating point emulation, and storage allocation alignment. Options controlling runtime error checking can be specified from within the source file or on the command line. The listing cannot be controlled from within the source file.

The BLS generates neither in-line 8087 floating-point code nor 80186/286 code. All floating-point operations are emulated, and upgrades are available for the performance conscious. The generated code is tight, but some peephole optimization with Boolean inversion on testing might reduce the incidence of redundant instructions. Table 1 shows benchmark timings for code generated by the compiler.

The linker generates the LOD (Logitech execute) files and can produce overlays. An optional utility is available to translate LOD files to .EXE format. The MAP file produced by the linker does not give relative addresses of procedures within modules, which limits the use of SYMDEB or other generic debuggers. These addresses can be found in the source listing.

The libraries supplied with BLS form the basis of a complete systems programming package. DOS services, floating-point numerical routines, primitive concurrent programming, and several levels of file I/O are provided. All are well detailed in the documentation. Graphics routines are absent. A SetJump/LongJump utility similar to that found in C implementations would aid in logically handling errors (no GOTO statement is available in Modula-2).

The well-indexed user's manual has recently been rewritten, expanded, and typeset with a laser printer, although about 30 percent of the volume, mostly descriptions of library modules, is still in Courier monospace. It is now perfect-bound rather than loose-leaf, which will make documentation upgrades messy. It is thorough and contains detailed information about all of Logitech's software components, even those not included with the BLS. The "Getting Started" section includes good installation instructions and useful tutorials in both the BLS and the Modula-2 language itself.

The list of compiler errors does not expand on the probable cause of an error, and several numeric error codes have no text explanation at all.

Five environment string variables are available to control a search by the compiler and linker to locate files. In the recommended hard-disk configuration, these 5 variables use a total of 80 characters of this very limited DOS resource, leaving most users critically short. This problem can be avoided by upgrading to DOS 3.1. Other solutions include abandoning the suggested structure and imposing an original strucutre that uses fewer characters, or building batch files to reconfigure the environment strings to the task at hand.

The limits to the BLS compiler and linker are not restrictive. Options and upgrades are available from Logitech to produce ROMable and 8087 in-line code. Several useful tools are available, including two debuggers. A MAKE utility is not available but would be useful.

The low price of the Logitech Modula-2/86 BLS is not a reflection of the system's usefulness. The package is an excellent introduction into a sophisticated programming environment.

—JOHN T. COCKERHAM

MIX C

MIX Software
2116 E Arapaho Drive, Suite 363,
Richardson, TX 75081
800/523-9520; in Texas,
214/783-6001

PRICE: $39.95

CIRCLE 343 ON READER SERVICE CARD

Although MIX C version 2.0.2 from MIX Software is a full implementation of the C language standard as specified by Kernighan and Ritchie, it cannot be considered a professional-level compiler. The programs it produces are very slow—up to 10 times slower than those produced by Microsoft C. But at $39.95, MIX C is less expensive than most C compilers and, therefore, intended for use as a learning tool or for developing personal applications.

MIX C is available in versions for IBM PC and CP/M 80 systems. The compiler's specifications are shown in table 1. Version 2.0.2 offers structure assignment and enumerations. Only a single memory model, called .COM, is supported, because a single 64KB segment (a DOS .COM file) holds the code and the data for the program. A typical small memory model allows 64KB each for the code and the data.

As table 2 shows, MIX C offers basic facilities for compiling and linking C programs. The compiler's command-line switches do not take wild cards or multiple files on the command line, which is an inconvenience when working with multifile programs. Mix C's only switch allows the user to redirect the listing output to a file or a printer. As with Pascal, additional compiler options are specified by embedding specially formatted comments in the source code. The compiler can print an attractively formatted source listing showing **#include** files and macro expansions.

MIX C's linker uses a proprietary object file format and does not accept typical DOS .OBJ files unless they are first converted by a separate utility program. The linker can generate overlays to help squeeze large programs into the 64KB memory mode.

A conversion utility called .ASM Utility that can prepare MASM and non-MIX C object files for the MIX C linker is available for an additional $10. The utility's features are documented scantily in a READ.ME file. Even so, add-on libraries for the MIX C compiler are not readily available; the utility does not allow conversion.

Overall, MIX C's documentation ranges from poor (for technical information) to good (for general information and the tutorial sections). Various

TABLE 1: *Specifications*

Version tested	2.0.2
Supported on other systems	See text
Cross-compiler hosts	No
Availability of add-on libraries	No
Minimum disk space required	200KB
Minimum RAM	128KB
Supports full language	Yes
Full standard library	Yes
PC-specific functions	Yes
Assembly language interface	Yes
COMPATIBILITY	
MASM	$10
LINK	No
SOURCE CODE	
Start-up sequence	No
Library functions	Yes
MEMORY MODELS	
Large	No
Medium	No
Compact	No
Small	No
.COM	Yes
OTHER PROGRAMS INCLUDED	
Librarian	No
Assembler	No
Linker	Yes
Source-level debugger	No
MAKE	No
Other	Edit, SPEED-UP, etc.

These specifications can be compared with those for other C compilers in table 1 in "The State of C" (William J. Hunt, January 1986, p. 84). See also table 1 in "Whitesmith's C Compiler," (Product Watch, Marty Franz, June 1986, p. 201) and table 1 in "Let's C," (Product Watch, Marty Franz, August 1986, p. 177).

The MIX C compiler requires only half as much disk space as the smallest compiler reivewed in "The State of C," (William J. Hunt, January 1986, p. 82). Unfortunately, user programs must also be small; only the .COM memory model is supported.

TABLE 2: *Compiler Functions*

COMPILER OPERATION	
Single-step compile command	Yes
Compile and link	No
Accepts lists of files	No
Accepts wild cards	No
Lists preprocessor output	Yes
Lists assembler output	No
Line numbers in error messages	Yes
Header file search list	See text
Flexible disk file layout	See text
C LANGUAGE EXTENSIONS	
Embedded assembly language	No
Void function returns	No
Enumerated types	Yes
Structure assignment, etc.	Yes
Function argument checking	No
LIBRARY EXTENSIONS	
Math functions (sqrt, exp, etc.)	Yes
Unbuffered file I/O	Yes
Keyboard input (low-level)	Yes
PC screen output (cursor control, cursor attributes, scroll)	Some
Execute programs/DOS (exec/fork and system)	Yes
DOS services (date, time, etc.)	Some
PC-specific functions	Some
UNIX-compatible functions	Yes
Error recovery (setjmp ( ), longjmp ( ))	No
FILE I/O	
Redirection	Yes
Full path names	Yes
DOS 1.1 support	No
DOS 3.1 file sharing	No
Record locking	No
ASCII/binary mode	Yes
MEMORY USAGE	
Overlays	Yes
Default stack size	Yes
Stack size settable	Yes
Stack overflow checking	Yes
8086 FAMILY SUPPORT	
Byte/word alignment	No
80186/80286 support	No
8087/80287 support	No
Automatic sensing	No
ROM support	No

These compiler features can be compared with those for other C compilers in table 2 in "The State of C" (William J. Hunt, January 1986, p. 86). See also table 2 in "Whitesmith's C Compiler," (Product Watch, Marty Franz, June 1986, p. 202) and table 2 in "Let's C," (Product Watch, Marty Franz, August 1986, p. 178).

MIX C offers only basic compile and link options. The compiler accepts no lists or wild cards on the command line.

TABLE 3: *Documentation Quality*

INSTALLATION	
Packing list	Yes
File inventory	Yes
Key files described	Yes
Quick step-by-step procedure	Yes
Instructions for floppy and hard-disk configurations	Yes
List changes from last version	Yes
SET-UP	
Set-up assumptions described	Good
Notes on RAM/second hard disk	Poor
OPERATIONS EXPLAINED	
Compile options	Good
Compiler error messages	Good
Linking C programs	Good
Runtime error messages	Good
Runtime options	Fair
LANGUAGE/LIBRARY SPECIFICATIONS	
Deviations from Kernighan and Ritchie standard	Good
Data type representation	Good
Memory models and memory layout	Fair
DOS and PC-specific features	Fair
ASSEMBLY LANGUAGE INTERFACE	
Segment, group, and class specification	Poor
Standard prologue, epilogue	Poor
Instruction formats for args, public, extern, struct	Poor
Return value conventions	Poor
Complete examples	Poor
FILE I/O	
Redirection	Fair
Console I/O	Good
Device I/O	Good
Buffered versus unbuffered	Good
ASCII versus binary modes	Good
LIBRARY DOCUMENTATION	
Average lines per function	25
Cross-reference information	No
Functions in table of contents	Fair
Examples of use	Fair
MANUAL ORGANIZATION	
Detailed table of contents	Fair
Index with functional entries	Fair
Order of function documentation	Sec.
OVERALL RATING	Fair

These notes on the documentaion can be compared with those for other compilers listed in table 3 in "The State of C" (William J. Hunt, January 1986, p. 88). See also table 3 in "Whitesmith's C Compiler," (Product Watch, Marty Franz, June 1986, p. 203) and table 3 in "Let's C," (Product Watch, Marty Franz, August 1986, p. 178).

The documentation's introduction and tutorial are well done; the technical details, however, are weak.

aspects of the documentation are rated in table 3. The single, soft-bound manual contains a lengthy tutorial section that introduces the novice C programmer to the language through small programs that can be compiled and linked directly using the MIX C package. While not a very complete introduction, the tutorial acts as a good companion to Kernighan and Ritchie's book. (Other books on the C language are reviewed in this issue in "C Companions," Book Reviews, p. 199.)

Installation of the compiler is straightforward. However, inexperienced users may have difficulty changing the default library and **#include** file directories, which requires patching the compiler and linker .COM files using DEBUG. The documentation for this procedure is contained only in a READ.ME file on the disk. This is an unacceptable procedure for a product otherwise best suited to new users.

TABLE 4: *Benchmarks*

COMPILE TIMES	
60-line file	11.9
150-line file	27.3
500-line file	59.1
LINK TIMES	
1 object file	17.0
6 object files	19.3
PROGRAM SIZES (bytes)	
Eratosthenes sieve	29,069
Pentathlon	30,084
GENERAL OPERATIONS	
Function calls (Fibonacci)	104.9
Integer arithmetic	40.0
Long arithmetic	269.9
Subscripts (character count)	42.3
Pointer use (string copy)	88.9
With register variables	88.9
Eratosthenes sieve	28.7
With register variables	28.7
FILE I/O	
Read/write	
Floppy disk to floppy disk	43.4
Hard disk to hard disk	18.1
Getc/putc	
Floppy disk to floppy disk	110.6
Hard disk to hard disk	64.3
Floating-point operations	
Add/multiply (dot product)	96.9
Exp/log	155.2
Sin/tan (trig functions)	196.0

Times are shown for the compact model only.
Benchmarks were run on an NCR PC6 running DOS 2.11 with a 20MB hard disk, 640KB of memory, and a single floppy disk. The times can be compared with those for other C compilers in table 4 in "The State of C" (William J. Hunt, January 1986, p. 90). See also table 4 in "Whitesmith's C Compiler," (Product Watch, Marty Franz, June 1986, p. 205) and table 4 in "Let's C," (Product Watch, Marty Franz, August 1986, p.179).

The MIX C utility SPEEDUP slashed execution times by a factor of ten. However, even after the SPEEDUP process, MIX C's times still were much slower than those for the compilers reviewed in "The State of C," (William J. Hunt, January 1986, p. 82).

Table 4 shows the results of the performance benchmarks, run on an NCR PC6 (Norton Utilities SI Index: 1.0) using DOS 2.11, a 20MB hard disk, a Princeton Colorview graphics card, 640KB of memory, and a single floppy disk. The module sizes are obtained by adding the .COM file produced by the linker to the size of the runtime overlay file RUNTIME.OVY, required for execution. A smaller version of the runtime overlay is supplied for programs that require additional memory; a program called SHRINK can reduce the size of the object modules themselves.

As the benchmarks show, MIX C performs adequately in integer and pointer applications. However, long-integer, floating-point, and character I/O applications are especially slow. The compiler uses binary-coded decimal (BCD) math for floating-point numbers, which eliminates round-off errors, but exacts a performance penalty.

A program called SPEEDUP was run on the object files before the benchmark times were taken. This increased MIX C's performance to as much as 10 times faster than that of the original, unoptimized versions.

Despite the low price, users should think twice before purchasing MIX C. The programs generated by the compiler perform slowly. The 64KB memory model, the lack of add-on libraries, and numerous quirks (limited compilation options and using SPEEDUP to optimize the programs) combine to make MIX C unsuitable for professional use.

—MARTY FRANZ

The AI Challenge

Emulating the human thought process has proven an arduous task, but AI techniques are nonetheless influencing the development of products for the PC.

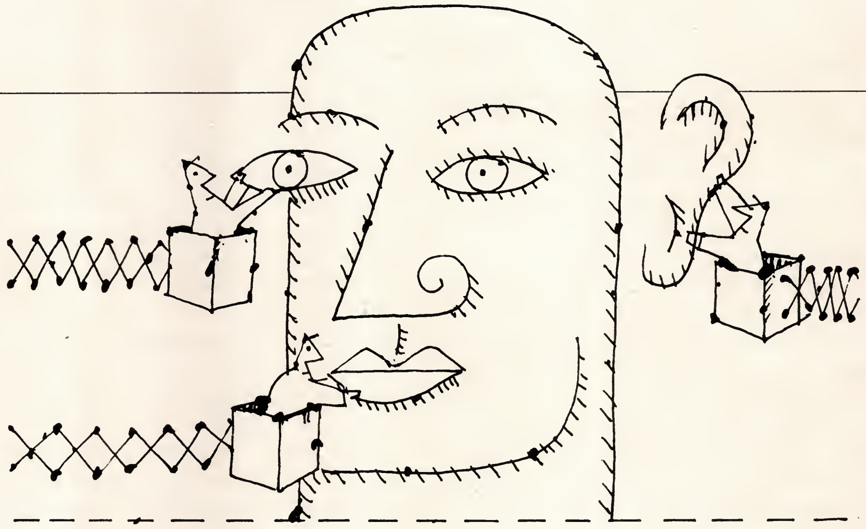


ILLUSTRATION • MACIEK ALBRECHT

Artificial intelligence is definitely in vogue. Many new products tout its use, and just about every software company is working on future products incorporating it. This month we give a general introduction to the topic.

The field of artificial intelligence can be loosely defined as the study of techniques allowing machines to exhibit some manner of human-like intelligence in action or conclusion. Perhaps the most important lesson that AI researchers have learned over the last few decades is how difficult it is to emulate or reproduce the effect of human thinking. When the field became popular in the 1960s, some researchers envisioned that humanoid butlers, equally conversant in Russian and English, would grace the Sears catalogs of the 1980s. While such notions seem naive in retrospect, remember that computer science technology in general was developing (as it still is) at an exponential rate. It seemed quite reasonable at the time to expect machine reasoning, automatic language translation and understanding, and many other elements of AI to be developed at a similar rate.

The fact that, by and large, AI has not evolved as fast as predicted is more a testimony to human ability than it is a criticism of AI research. Even the simplest of human activities, such as plotting a course to the refrigerator, requires both a sophisticated model of the real world and the ability to process enormous quantities of input data.

This does not mean that AI's challenges are too difficult to surmount, but only that patience is required. By nibbling away at specific aspects of AI problems rather than looking for general solutions, researchers have accomplished a great deal. Progress in the area of expert systems, for example, has been made possible by focusing on narrow problem areas, such as specialized medical diagnosis or lunar geology. Successes in robotics have resulted from concentrating on problems of industrial automation rather than pursuing the six-million-dollar-man or a general-purpose robot.

Some of the most important contributions have been made not in the area of applications, but rather in tools. AI languages such as LISP and Prolog, first understood and used only within the cloisters of academic research, are now available commercially. (*PC Tech Journal* has published reviews of several such products. See "Creating a Standard LISP," Mark Bridger and John Frampton, December 1985, p. 98 and "Programming in Logic," Michael Covington, January 1986, p. 145.)

In just the last few years, AI also has crept into software applications that historically have incorporated relatively mundane computer technology. Databases, word processors, and spreadsheet programs are now available that use, to varying degrees, techniques originally developed within the AI research community. The emphasis in these applications is on simplifying user interfaces. Computer programming languages and applications have tended to interact with the user at a level of conceptualization closer to how the machine works than to how the user thinks about the problem. The user has been forced to deal with the machine on the machine's terms. The new generation of programs seeks to elevate the level of user interaction, giving to the program the burden of devising machine-level steps for problem solving. This may take the form of streamlining user assistance to require less computer-specific knowledge from users or to allow them to communicate instructions in a more natural language.

AI TECHNIQUES

While humans are able to understand an English sentence or find their way through a building without giving it conscious thought, the human brain actually must sort through a voluminous quantity of information in selecting the appropriate response. The amount of extraneous information that must be ignored during the decision-making process is staggering.

AI presents different methods for reproducing this amazing human ability to sort through the irrelevant. As of yet AI has developed no single unified theory, only a loose collection of techniques and concepts.

The way in which knowledge is represented and understood within a machine can have a profound effect on the machine's usefulness. The millions of bits per second of data received by a video sensor on a robot, for example,

are worthless unless they can be integrated and represented internally in a meaningful way. The information must be stored so that the program can rapidly determine whether new data support or rule out possible interpretations. Relevant details must be separated quickly to help process future user input. Researchers have devised several different approaches for organizing information, described below.

Frames. Frames organize information around expected scenarios. These scenarios are abstract descriptions of typical events, motivations for the event, players involved—all cues for recognizing that a given situation is occurring and for establishing all relevant details. A birthday party, for example, is known to consist of a birthday celebrant and guests, typically with a birthday cake, candles, and gifts for the celebrant. This high-level organization for an entire context aids in knowing which questions to ask the user or how to understand future user input.

Semantic networks. Pointers from one fact to other related facts can be organized in semantic networks to establish relationships between certain items of information. Representing knowledge about a family might include a pointer from a "John" node to a "Mary" node labeled "is-son-of."

Semantic rules. Knowledge also can be organized with rules that take the form of unconditional declarations of information, such as "John is Harry's boss," or conditional information, such as "If X is Y's boss, then Y is X's subordinate."

The collection of semantic rules forms the information base, which can be used for different purposes: deduction, synthesis, and planning. *Deduction* involves reaching conclusions about whether certain facts follow from what is known. Given a tentative conclusion, deductive techniques are used to determine that the conclusion is plausible—consistent with known information—or is implied by known information. Given the information that "if today is Tuesday, this must be Belgium," then "being in Belgium" is plausible. Of course, this would be true only if no other specific information was known about location, such as "being in France." If "today is Tuesday," is also known, then "being in Belgium" is implied.

Synthesis of information to reach new conclusions can take different forms. Additional useful information can be inferred that either is directly implied by known information or is a plausible generalization. Such a tentative generalization can be of working assistance until disproven. Knowing that a car is parked at Mary's house every night and that Mary drives the car every day to work might prompt the conjecture that Mary owns the car.

Planning uses current knowledge to determine a sequence of steps needed to accomplish a specific goal. This can involve deduction and synthesis to determine a chain of actions that will lead from the current context to the desired goal. Typically, planning means sifting through an enormous set of possible choices, the overwhelming majority of which would not be effective. *Heuristics* must be used to determine the best steps to be taken at each point. Heuristics are rules that may be derived from hunches, commonly practiced guidelines, or perhaps what has or has not worked in the past. A heuristic for the game of Tic-Tac-Toe, for example, might be always to take the center square if it is available.

AI LANGUAGES

Various languages have been developed within the AI community to facilitate the techniques just discussed. AI languages are not prerequisites to the use of AI techniques, however. The two most popular AI languages, LISP and Prolog, provide higher-level support for some aspect of AI programming.

LISP, originated at MIT in the mid-1960s, is based on the general concept of a list. The language represents both programs and data as lists of objects. Each object can itself be a list. Primitive functions operate on lists of items and manage *association lists*; these list arbitrary, named objects for which LISP provides built-in functions to retrieve and update rapidly. The primary mode of programming is through the composition of recursive functions. Its generality of structure, coupled with a comprehensive programming environment, have given LISP widespread acceptance within the AI community.

Prolog is a logic programming language developed at the University of Marseilles in 1972. Programs in Prolog are a collection of logical rules, such as "Socrates is human" or "X is fallible if X is human." A program is executed when the user proposes a tentative conclusion, such as "Someone is fallible" or "Socrates is fallible," for Prolog to prove. The language uses built-in pattern matching and backtracking strategies to determine which rules are applicable to the problem solution.

AI APPLICATIONS

AI techniques have been applied in a wide variety of areas. Several of these are discussed below.

Vision understanding. Vision understanding entails analyzing digitized input from an external scene and recognizing common objects and contexts. AI techniques reason about the identity of an object from incomplete information owing to hidden structure, imperfect information, or variability from standard characteristics of the object.

Robotics. Applications involving robotics include industrial process control, automated vehicle movement, and automated assembly. Attempting to automate control of equipment or motion may involve analyzing the scene to understand the current context, then planning a sequence of steps consistent with environmental constraints.

Expert systems. Expert systems take a particular application area and attempt to play the role of an expert in making or assisting with decisions.

Knowledge engineering. The process of synthesizing a set of rules and constraints to govern decision making is known as knowledge engineering. The rules are typically culled from discussions with experts in that field. The rules are then embedded in an automated reasoning system that plans a course of action to solve a specific problem, given information made available to it. The key to success is finding narrow application areas with a small number of succinctly stated rules that govern decision making. Two very successful early examples are the MYCIN system to aid physicians in diagnosing and treating meningitis and other bacterial infections and the DENDRAL system for examining a spectroscopic analysis of an unknown molecule to isolate embedded molecular structure.

Knowledge-based assistants. Some expert systems are embedded in products such as databases, spreadsheets, word processors, or electronic mail handlers to watch and learn a user's pattern of usage. These are knowledge-based assistants, sometimes known as mixed initiative systems because input is provided by both the user and the system, as appropriate. The information collected by these systems can be used to watch for certain kinds of errors, fill in details automatically when patterns recur, or offer additional explanation for system actions.

Higher-level interfaces. By using higher-level interfaces for programming or application areas such as databases or word processing, the user model for all interaction is elevated to hide certain details completely.

Natural language. English as a communications medium between system and user was one of the original focuses for AI research. Critical to the effectiveness of such natural language systems is circumscribing the system's ability to understand vocabulary and sentence structure. This includes entering into a dialog with the user to resolve any misunderstanding or ambiguity.

Program sythesis. Another use of AI to provide a higher-level system interface is known as program synthesis. The user states a desired result, without giving any indication of how to achieve it. This statement could be phrased as a simple request, an example of desired outcome, or an analogy to understood behavior. It could be stated in English or illustrated visually.

Program verification. Proving properties in mathematics was an early application of automated reasoning techniques. A more recent use is to establish certain properties of computer programs, called program verification, or program correctness. The user writes a formal mathematical statement of what it means for a program to "work," then interacts with a program verifier in order to establish that all executions of the program indeed will work as intended. Program verification is a complement to program testing methods, which show that a program works correctly for specific executions.

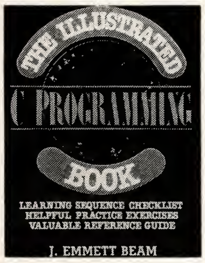
In the last two years, AI-related products have begun to reach the personal computer marketplace; many more such products are now under development. Commercial products incorporating AI techniques and tools for developing AI programs are available for the IBM PC, as are several implementations of LISP and Prolog. We are on the cusp of having the speed and memory on personal computers necessary to develop significant applications employing AI techniques.

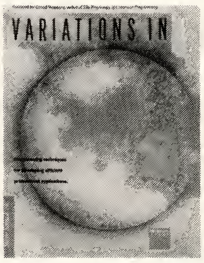
In future columns, we will try to shed some light on what AI is and is not. We will discuss what tangible benefits can be expected from current and future applications of AI. Many of these benefits will have a direct impact on those using and developing PC products of the future.

Richard Schwartz, Ph.D., and Robert Shostak, Ph.D., are vice presidents of software development and cofounders of Ansa Software.

C Companions

Four books about programming in C attempt to pick up where Kernighan and Ritchie left off.





The Illustrated C Programming Book
J. Emmett Beam (Wordware Publishing, Plano, TX) 286 pages, paper, $19.95

Common C Functions
Kim Jon Brand (Que Corporation, Indianapolis, IN) 275 pages, paper, $17.95

C Wizard's Programming Reference
W. David Schwaderer (Wiley Press, New York, NY) 200 pages, paper, $19.95

Variations in C
Steve Schustack (Microsoft Press, Bellevue, WA) 368 pages, paper, $19.95

When it was first published in 1978, Kernighan and Ritchie's *The C Programming Language* (Prentice-Hall) was an obscure book about an obscure language. Since C became available for microcomputers, the book has become the accepted standard C reference.

Users of C must have a good computer science background to understand the concepts presented. While concise and accurate, the book does not provide many examples and programs. Several pitfalls of programming in C are mentioned only briefly.

Publishers have filled in the gaps left by Kernighan and Ritchie, offering books ranging from those that cater to the newcomer to those intended for the advanced C programmer. Four such books are reviewed here.

The Illustrated C Programming Book is a set of programmed-instruction (PI) exercises in C. It begins with an overview of a C program, then proceeds through 90 modules to the concept of program portability. All statements, functions, and concepts needed to write C programs are included.

The book features helpful box diagrams, flowcharts, and code samples. One of the major problems with this book is that it lacks compiler-specific information. Further, the portability module covers only some of the most basic pitfalls that are encountered when converting programs from one compiler to another (often a major task for C programmers working on an IBM PC). Also, many of the little annoyances of programming in C are dealt with only in the examples, not in the text. On the whole, however, *The Illustrated C Programming Book* is a valuable source for someone with the patience to work through a PI textbook. It might make a good companion for a C interpreter user (see "The State of C Interpreters," Marty Franz, May 1986, p. 153).

Common C Functions works on the theory that, by compiling the functions, then writing programs that use them, the novice will learn to program in C. The book is organized into two sections. The first presents a summary of C programming techniques. The second is a set of C utility functions that includes input handling, data conversion, and report formatting, as well as four executable sample programs.The book's main drawback is that not all functions presented will compile and run on all C compilers.

Intended as a reference book and a companion to Kernighan and Ritchie, *The C Wizard's Programming Reference* contains tutorial information not covered in *The C Programming Language.* Clear, complete descriptions are included of topics often glossed over in other books. The encyclopedic format is probably more useful to experienced C programmers than beginners, but even novices will find the book handy. "Hazard" paragraphs scattered throughout the text warn of potential pitfalls.

Variations in C covers the process of writing production-quality programs in C. Published by Microsoft Press, the striking typography and page layout of this book are beautiful. Important points throughout the text are highlighted in boxes, and a clear monospace font is used for all the listings. The content is excellent. After covering the basics, the author deals with the more esoteric points of programming in C, such as coding style, semantic typing, and overall program organization.

Intended specifically for use with Microsoft C 3.0, the book is only partially applicable to other C compilers. The rigorous coding style necessary for large projects is developed early in the book and followed carefully through all the source code presented. Programmers working on large development projects definitely will want to consult *Variations in C* before beginning their work. Some beginners may find this book's content too advanced.

Of the four books reviewed, only *The C Wizard's Programming Reference* is sufficiently general to be a useful companion to C programmers at all levels of experience. *The Illustrated C Programming Book* and *Common C Functions* primarily address the needs of novice C programmers, while *Variations in C* is best suited for the advanced programmer. Each of these four books supplements the work begun by Kernighan and Ritchie.

—MARTY FRANZ

INDEX TO PRODUCTS

Use these reader service cards to get FREE INFORMATION about the products and services in this issue of TECH JOURNAL

Learning more about a product that's advertised or mentioned in an article in this month's issue is as simple as 1-2-3. And absolutely free.

1 Print or type your name and address on the attached card. Use only one card per person.

2 Circle the numbers on the card that correspond to the numbers at the bottom of the advertisements or articles for which you want more information.

(Key numbers for advertised products also appear in the Advertisers' index.)

3 Simply mail the card, and the literature will be mailed to you free of charge by the manufacturer.

TECH PC JOURNAL

Are you personally involved in the selection of microcomputers and related products for:

1 Your company or organization?
A☐ Yes B☐ No

2 Your client companies or organizations?
C☐ Yes D☐ No

3 Are you planning to purchase in the next 6 months:
E☐ PC Hardware?
F☐ PC Software?
G☐ PC Peripherals?

101	116	131	146	161	176	191	206	221	236	251	266	281	296	311	326	341	356	371	386
102	117	132	147	162	177	192	207	222	237	252	267	282	297	312	327	342	357	372	387
103	118	133	148	163	178	193	208	223	238	253	268	283	298	313	328	343	358	373	388
104	119	134	149	164	179	194	209	224	239	254	269	284	299	314	329	344	359	374	389
105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390
106	121	136	151	166	181	196	211	226	241	256	271	286	301	316	331	346	361	376	391
107	122	137	152	167	182	197	212	227	242	257	272	287	302	317	332	347	362	377	392
108	123	138	153	168	183	198	213	228	243	258	273	288	303	318	333	348	363	378	393
109	124	139	154	169	184	199	214	229	244	259	274	289	304	319	334	349	364	379	394
110	125	140	155	170	185	200	215	230	245	260	275	290	305	320	335	350	365	380	395
111	126	141	156	171	186	201	216	231	246	261	276	291	306	321	336	351	366	381	396
112	127	142	157	172	187	202	217	232	247	262	277	292	307	322	337	352	367	382	397
113	128	143	158	173	188	203	218	233	248	263	278	293	308	323	338	353	368	383	398
114	129	144	159	174	189	204	219	234	249	264	279	294	309	324	339	354	369	384	399
115	130	145	160	175	190	205	220	235	250	265	280	295	310	325	340	355	370	385	400

Please print clearly—Use only one card per person. Void after December 31, 1986

Name__________ Phone (____)__________

Title__________

Company__________

Address__________ Apt.____

City__________ State______ Zip______

(Zip code must be included to insure delivery.)

4☐ **Please send me 1 year (12 issues) of *PC Tech Journal* for $23.70 and bill me. I'll save 50% off the single copy price!**

TJ9862

TECH PC JOURNAL

Are you personally involved in the selection of microcomputers and related products for:

1 Your company or organization?
A☐ Yes B☐ No

2 Your client companies or organizations?
C☐ Yes D☐ No

3 Are you planning to purchase in the next 6 months:
E☐ PC Hardware?
F☐ PC Software?
G☐ PC Peripherals?

101	116	131	146	161	176	191	206	221	236	251	266	281	296	311	326	341	356	371	386
102	117	132	147	162	177	192	207	222	237	252	267	282	297	312	327	342	357	372	387
103	118	133	148	163	178	193	208	223	238	253	268	283	298	313	328	343	358	373	388
104	119	134	149	164	179	194	209	224	239	254	269	284	299	314	329	344	359	374	389
105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390
106	121	136	151	166	181	196	211	226	241	256	271	286	301	316	331	346	361	376	391
107	122	137	152	167	182	197	212	227	242	257	272	287	302	317	332	347	362	377	392
108	123	138	153	168	183	198	213	228	243	258	273	288	303	318	333	348	363	378	393
109	124	139	154	169	184	199	214	229	244	259	274	289	304	319	334	349	364	379	394
110	125	140	155	170	185	200	215	230	245	260	275	290	305	320	335	350	365	380	395
111	126	141	156	171	186	201	216	231	246	261	276	291	306	321	336	351	366	381	396
112	127	142	157	172	187	202	217	232	247	262	277	292	307	322	337	352	367	382	397
113	128	143	158	173	188	203	218	233	248	263	278	293	308	323	338	353	368	383	398
114	129	144	159	174	189	204	219	234	249	264	279	294	309	324	339	354	369	384	399
115	130	145	160	175	190	205	220	235	250	265	280	295	310	325	340	355	370	385	400

Please print clearly—Use only one card per person. Void after December 31, 1986

Name__________

Title__________ Phone (____)__________

Company__________

Address__________ Apt.____

City__________ State______ Zip______

(Zip code must be included to insure delivery.)

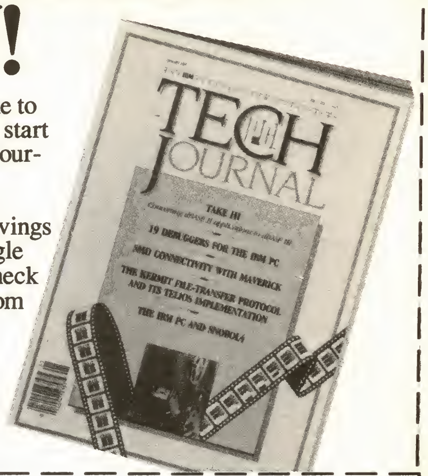
4☐ **Please send me 1 year (12 issues) of *PC Tech Journal* for $23.70 and bill me. I'll save 50% off the single copy price!**

TJ9861

FREE INFORMATION

Follow the instructions
on the reverse side of
this card to
receive advertisers'
product information.
FREE.

SUBSCRIBE NOW!



Now's an ideal time to consider having us start you as a PC Tech Journal subscriber.
12 issues cost you only $23.70...a savings of 50% off the single copy price. Just check the box at the bottom of the reply card.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

Postage will be paid by addressee

TECH JOURNAL
P.O. Box 40086
Philadelphia, PA 19106-9931

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

Postage will be paid by addressee

TECH JOURNAL
P.O. Box 40086
Philadelphia, PA 19106-9931

NEED IT FAST? WANT TO KNOW MORE? DON'T HAVE TIME TO SHOP AROUND?

INTRODUCING...

If you're like most system professionals you're up-to-date about the products in the PC marketplace. You're aware of brand and model differences, are informed about connectivity and compatibility problems, and you shop for competitive prices and fast service.

You're also probably among the many PC TECH JOURNAL readers who purchase by mail. That's why we're starting THE MART—PC TECH JOURNAL'S First Class Mail-Order Section.

Starting this month, and every month hereafter, you'll find the products you're looking for advertised in THE MART—and you'll benefit from the fast service and helpful support that identifies PC TECH JOURNAL advertisers.

If you're ready for First Class service, you're ready for THE MART.

MOVING?

Please write to: PC TECH JOURNAL Magazine, P.O. Box 2968, Boulder, CO 80322.

Include your mailing label from a recent issue of PC TECH JOURNAL for faster service. Please allow up to 60 days for change of address to take place.

COMPUTING POWER

FROM FLOPPY DISK SERVICES

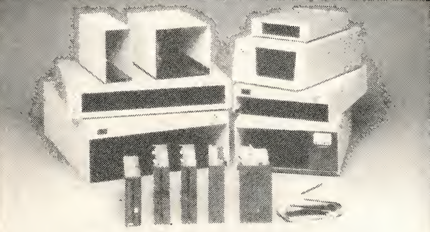
PRICES AND SPECS SUBJECT TO CHANGE WITHOUT NOTICE.

Floppy Disk Services has been supplying storage systems to the hobby and professional computer community for 7 years now. You can buy with confidence from a supplier that is in for the long haul. Whether you need advice or technical help, our staff is here to serve you.

The 'Service' in Floppy Disk Services stands for the confidence you'll have in dealing with us. We do not make you wait when a problem comes up. We will be there to help, and more importantly support and guide you. In the unlikely event we cannot answer your questions, a staff member will return a call to you in a timely and professional manner with the answer.

Warranty is a term that sometimes is taken for granted. At Floppy Disk Services, we support one of the strongest warranty policies in the business. Our policy is to replace any equipment found to be defective in any way during the warranty period. Its as simple as that! No waiting for the merchandise to be sent back to the factory. If we find a problem, (other than abuse), we simply ship a new unit back to you.

The XT-clone system pictured above is our Super System VII XT clone. It contains a turbo processor, a 20mb hard disk formatted RLL to 31mb, 640K RAM, two 360K floppies, 1 AT 1.2 mb drive, mono amber monitor, par port, ser port, clock and AT type keyboard! You might expect to pay thousands for this system, but Floppy Disk Services, inc. will supply it ready to run with a 1 year warranty. Call for latest quotes on your custom system needs. . .



Complete line of drive enclosures

800-223-0306

CALL FOR FREE CATALOG

Dealer Inquiries Invited

VISA MasterCard

39 Everett Dr.,Bldg. D Lawrenceville, NJ 08648

(609) 799-4440

FLOPPY DISK SERVICES INC.

INDEX TO ADVERTISERS

TECH MART

ATTENTION SOFTWARE DEVELOPERS!
More Users For Less Money!
Digitrol's software solution
to multi-user.

pcSHARE
MULTI-USER O/S

pcSHARE allows your IBM-XT/AT or compatible to support up to 5 users running your vertical market programs and 1-2-3, dBASE, WordStar, and more on inexpensive serial CRTs. pcSHARE efficiently runs compiled Basic, dBASE, Pascal & C programs with full DOS 3.0 compatible record locking.

DIGITROL COMPUTERS INC.
440 Phillip Street,
Waterloo Ontario, CAN N2L 5R9
(519) 884-4541

IBM XT/AT, Lotus 1-2-3, dBASE, and WordStar are trademarks of IBM Corp. Lotus Development Corp., Aston-Tate and MicroPro International respectively.

CIRCLE 375 ON READER SERVICE CARD

64K ■ 128K ■ 256K
DRAMS
80287-8 ■ 80287-3
8087-3 ■ 8087-2
8087-1

BITTNER
ELECTRONICS

899 SOUTH COAST HIGHWAY
LAGUNA BEACH, CA 92651
(714) 497-6200
CALL NOW FOR **FREE** CATALOG

CIRCLE 376 ON READER SERVICE CARD

SUPER LOCK™

IBM PC, XT and AT programmers:

Don't give away results of your effort!
Don't waste your money buying costly fingerprinted diskettes!

You can have your programs automatically protected against piracy!

* No special training needed!
* No confusing instructions!
* No discouraging procedures!

No Sales Tax on orders outside N.Y. State

$ 180

YETIWARE CORP.
P.O. BOX 21152
MID TOWN STATION
NEW YORK N.Y. 10129

DEALER INQUIRIES INVITED

CIRCLE 377 ON READER SERVICE CARD

Protect Your Private Parts.

Whether you're a software developer or an end user, whether you work alone on a PC or you're part of a 24-person project team, keeping new software or sensitive data from prying eyes is serious business.

Short of complicated, wizard-like protection schemes, or James Bond, you haven't had too many realistic alternatives. Until now.

Because Rainbow Technologies has developed simple, easy-to-use systems called Sentinels that say no to prying eyes. No matter how hard they look.

The key to Sentinel is just that—a smart piece of plastic smaller than a best-selling cassette that plugs into the parallel port of your IBM PC/XT/AT or compatible.

There are two ways to go. The Software Sentinel™, that prevents anyone from using your licensed software program unless you give them a key. And the Data Sentinel™, Rainbow Technologies' newest prying eye-stopper, that automatically protects sensitive personal or project data as you enter it. The Sentinel Security Systems from Rainbow Technologies. The only hard part is the key.

RAINBOW TECHNOLOGIES, INC
17971 Skypark Circle, Suite E, Irvine, CA 92714
Phone (714) 261-0228 Telex: 386078

CIRCLE 378 ON READER SERVICE CARD

PERFECT COPIES GUARANTEED

Why risk duplicating your important programs on your computer, when *our* equipment is designed solely to duplicate disks and verify their perfection 100%?

Over 600 formats. 3½", 5¼" and 8". Plus serialization, copy protection, labeling, packaging, shrink-wrapping and *fast,* personalized service.

Western Transdata Inc.
DISK DUPLICATION

1701 E. Edinger Ave.
Building A4
Santa Ana, California 92705
(714) 547-3383 (Collect)

CIRCLE 379 ON READER SERVICE CARD

NEW EPROM PROGRAMMER

EPROM 1
$495

*Programs all 28-Pin, 5-Volt EPROMs and EEPROMs.
*Perfect for engineering workstations, field service and small production runs.
*Simple keyboard and display for stand alone duplication.
*Interface to your computer via an RS232C port.
*PC control software included at no extra charge.

TO ORDER PHONE:
800-325-6028 Outside California
916-885-7262 Inside California
VISA AND MASTERCARD ACCEPTED

INTERNATIONAL MICROSYSTEMS INCORPORATED
11554 C Avenue, Auburn, CA 95603

CIRCLE 380 ON READER SERVICE CARD

ASMLIB

ASSEMBLY PROGRAMMING LIBRARY

- NO ROYALTIES REQUIRED
- Graphics - Color, Herc. Monochrome, EGA
- Floating Point Trigonometry and Arithmetics
- 8087 Supported but not required
- ASCII to Numeric conversions
- Formatted Output (PRINT USING)
- Console I/O with Windowing
- DOS Shells
- Sound Generation
- Plus much, much more

ASMLIB has over 170 functions for the IBM PC/XT/AT or compatible under DOS 2.00 or newer. Supplied in MS Assembly language source code on 3 DOS Diskettes with 215+ page reference manual.

All for only . . . $149.00 ppd

BC ASSOCIATES
13073 Springdale Street, Suite 134
Westminster, California 92683
For VISA, MC or COD, call (714) 741-3015

CIRCLE 382 ON READER SERVICE CARD

Get the whole story on graphics terminal emulation.

To find out more about software that lets your PC emulate TEKTRONIX™ 4105/6/7/9 and DEC VT100™ terminals, call or write:

GRAFPOINT
4340 Stevens Creeks Blvd., Suite 280,
San Jose, CA 95129 (408) 249-7951

CIRCLE 383 ON READER SERVICE CARD

TERMINAL EMULATION

Softerm PC emulates over 30 popular terminals including the:
- DEC VT102, VT220
- Data General D200, D410
- IBM 3101-20 (block mode)
- Hewlett-Packard 2622A
- Honeywell VIP7801, VIP7803

Guaranteed Compatibility
Call for free product brief
$195 MC-VISA-COD
For the IBM PC/XT/AT, DG1, NEC, Wang PC, TI Pro, Gridcase, Tandy

SOFTRONICS
7899 Lexington Dr., Ste 210
Colorado Springs, CO 80918
(303) 593-9540

CIRCLE 384 ON READER SERVICE CARD

TECH MART

TECH BOOK

Accessories/Supplies

BORLAND/ECOSOFT MANUALS HARD TO USE?
Convert your softbound manuals into IBM style looseleaf manuals. Kit includes everything you need to convert your manuals for easier use. We also have a Turbo Quick Reference Guide. Conversion kits $14.95, Quick Reference $10.95, plus shipping. Call now! VISA/MC
Computer Support Services
2607 78th Street
Lubbock, TX 79423
(806) 745-4050

Bar Coding

BAR CODE LABEL PRINTING
PrintBar II is a unique memory resident program that prints code 39 and UPC bar codes on Epson, IBM graphics and LaserJet printers. Print directly from almost any program, without copying data to special print files. 30 day money back guarantee. $295 plus $12 S/H second day air.
BEAR ROCK SOFTWARE CO.
P.O. Box 212
Placerville, CA 95667
(916) 622-4640

Business Opportunities

INCORPORATE!
*Save Taxes!
*Protect Your assets!
*Free Booklet!
*Free Forms!
*Save $— Do It Yourself!
*No Legal Fees!
WRITE OR CALL TOLL-FREE TODAY!
HARVARD BUSINESS SERVICES, INC.
3151 Summerset Rd.
Wilmington, DE 19810
800-345-CORP

Personal Computer Owners
CAN EARN $1,000 TO $5,000
monthly selling simple services performed by their computer. Work at home-in spare time. Get free list of 100 best services to offer. Write:
C.I.L.B.Q.
PO Box 60369
San Diego, CA 92106-8369

Hardware/Add-on Boards

Z80 Co-processors
For PC and PC/AT. From $199.50. Clock speeds to 9 Mhz. Run CP/M-80 software fast. Develop microprocessor software with our remote debugger kernel. Also run Intel ISIS software tools. We have Z80 processors for VAX and PDP-11 too.
Decmation
2065 Martin Ave #110
Santa Clara, CA 95050
(408) 980-1678

SPEECH SYNTHESIS
SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
1311 N. Main St.
Clawson, MI 48017
(313) 435-4222

FIXED DISK BIOS/BOOT
FiXT boots from most popular Hard Disks—DAVONG, TECMAR, IOMEGA, GT LAKES, etc. Adds XT-like BIOS interface to your disk for PC. Security, multiple volumes, removable media support optional. No-slot plug-in installation. Specify controller and computer with order. $80-$95. Add $3 shpg., CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue
Suite 201
San Diego, CA 92103
(619) 298-9349

PC-PROMPAC
ROM Expansion for PC!
Aldia systems introduces PC-PROMPAC, a "half-sized" PROM/ROM expansion board for IBM and IBM compatible PCs. PC-PROMPAC will support up to six 28-pin JEDEC compatible devices (ex: 2764, 27128, 27256, 27512, 6264, etc.) with individually selectable address ranges. Prices start at $125 for single units. Quantity discounts and OEM arrangements available. MC/VISA.
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, AZ 85069
(602) 886-1786

DIGITAL SIGNAL PROCESSOR
The Model 10 coprocessor board is based on the 16/32 bit TI TMS 32010 and is designed for applications in communications, speech, instrumentation, and numeric processing. A 1K complex FFT takes 90ms. Offered with onboard 12 bit 40 Khz A/D and D/A. Includes all utility and applications software. $650-$850.
Dalanco Spry
Suite 241 2900 Connecticut Ave. NW
Washington, DC 20008
(202) 232-7999

Hardware/Disk Drives

8 INCH DISKETTE SYSTEM
Read, write and format diskettes from IBM mainframes, minicomputers, data entry equipment, etc. Complete easy to use software handles EBCDIC conversion. Can read and write CP/M 8-inch diskettes (many formats). You also can use 8-inch drives for PCDOS files; 1200 KB per diskette! $1295 complete.
MicroTech Exports
223 Forest Avenue
Palo Alto, CA 94301
(415) 324-9114

Hardware/Peripherals

640K RAM UPGRADE
Easy installation. Absolutely no soldering or technical experience needed. This upgrade requires no slots; replacement chips plug into existing IC sockets. There are no software changes; simply install and use. 30 day money back guarantee. Specify IBM/XT or Compaq. Complete kit with illustrated instructions only $99, or $45 w/o RAM chips. Shipped UPS Blue ppd, foreign add $5. Visa, MC, Check/MO

Nuevo Electronics Corporation
P.O. Box 209, 111 South Second St.
Richmond, TX 77469
(713) 341-6001

TURBO START YOUR PC!!!
FOR IBM PC's ONLY!! The PC-INMATE, a Firmware ROM Module for the IBM-PC!!! Features: BOOT within 10 SECONDS!!! BREAK the 640K BARRIER!! Access up to 704K of RAM!!! (Lotus, DOS, etc.) SCREEN RELIEF-protects your monitor, and MORE!! Easily PLUGS IN-NO SLOT! Only $49.95 + $3 S/H. Dlr. inquiries accepted CK/MO/VISA/MC.
JS&J Software
1281 S. King St. Suite 6A
Honolulu, HI 96814
24 hr Orders Only: (800) 821-5226 ext. 435

IBM PC/XT/AT ...... SPEEDUP
*OVERTRUSTER: Boots the performance of PC and XT by 280%! 100% software compatible, is switchable between 4.77 and 7.38Mhz. $195.
*SPEEDSWITCHER: This is an economical 6 speed switch (4-11Mhz) for AT's. $89
*XCELEX: Automatic 5 speed switch (6-12Mhz) This unit will switch on the fly and works with ALL IBM AT's including the new Models with speed limiting Rom Bios. $149.
DATA MANAGEMENT SYSTEMS, INC.
1525 Hickory St.
Salina, KS 67401
ORDER HOTLINE (913) 823-6440

GM CAR DIAGNOSTICS TO PC
Connect the IBM PC to your GM car with our 85105 I/F and RS232. Our BASIC gives car trouble codes and sensor data from most 1980 ½ & newer cars. Exceptions include Cadillac. ($160) Store data on audio tape, with our 85104 cassette I/F. ($80) Play tape to 85104 to 85105 to PC.
EXECUTIVE ELECTRONICS INC.
938 Main Street
Yarmouthport, MA 02675
(617) 362-3694

CP/M & 1.2Mb AT ON PC
With MULTI-DISK card & UniForm-PC use 3.5, 5.25 & 8-inch single & double density CP/M format as DOS diskettes on your IBM PC or XT. Many MS-DOS formats supported including IBM AT 1.2 Mb. HP-150 & Data General 1. Over 200 formats. Both MULTI-DISK & Uniform-PC for $225. Disk drives & adapter cable available.
PS Engineering
P.O. Box 51068
San Jose, CA 95151-5068
1-800-369-2398; 1-800-423-7171 in CA.

EPROM/EEPROM PROGRAMMER
Programs 2716-27512, 25xx, 68764/66 eproms via RS-232. Also 874x micros, 28xxA & 52Bxx eeproms. Automatic Baud rate select, built in menus, no personality modules. Price: $250. Mention this ad for free terminal software. 16 BIT I/O MODULE $75
For control of input or output lines via RS-232. Use with modems for remote control.
INTELLITRONICS
P.O. Box 3263, Tustin, CA 92680
(714) 669-0614

Rates and Information

Standard listings consist of a bold lead line (25 characters maximum); 7 lines of body copy (45 characters per line); 4 lines for company name, address and telephone number. $140 per insertion—3 issue minimum. Additional charge for extra lines and company logos. Prepayment and frequency discounts available. American Express, MasterCard, Visa accepted. Copy subject to publishers approval. Send typewritten or printed copy, reproducible logo art (if applicable) and remittance to Kathryn Cumberlander, Classified Sales Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. For additional information, assistance, or to place an order by phone, call collect (212) 503-5115.

TECH BOOK

HARDWARE/ PERIPHERALS

PC SPEEDUP

Up to 45% speed improvement for IBM PC and clones. 100% compatible, easy installation. Kit includes NEC V20 or V30 Chip, Tool, User Guide, Warranty, free Multiuser BBS Access. V20-5 mhz for PC, V20-8 and V30 in stock. V30 replaces 8086. Call for Turbo 7 and VP/M.

From $4 to $89. Tech info 414-242-2165
Exec.-PC, Inc.
PO Box 11191 Shorewood, WI 53211 Order on BBS:
Orders: 414-242-2173 414-964-5160

DATA INPUT DEVICES

TPS provides Bar Code & Magnetic Stripe Readers for simple installation IBM PC, AT, 3161, 3163, 3164, 3191, 3194 terminals, as well as many other microcomputers and terminals. No card slot or RS-232 port is required, and the readers are transparent to all software. A bar code print program (code 39) is available for the PC & AT at only $50 with the purchase of a reader. A magnetic encoder is also available for the PC & AT.

TPS Electronics
4047 Transport Street
Palo Alto, CA 94303
(415) 856-6833

PUBLICATIONS/ CATALOGUES

Advanced TurboPascal Book

"Turbo Pascal—Advanced Applications" a new book for serious programmers. Written by the TP experts, it covers topics such as optimization techniques, interrupts, system level tools, graphics, and more. In-depth and thorough. $16.95; or with MS DOS disk $29.95. Add $1.50 shipping (US & Canada). Free info.

Rockland Publishing
190 Sullivan Crossroad, Suite 107
Columbia Falls, MT 59912
(406) 257-9119

SOFTWARE/BUSINESS

The Andsor Collection™

Unique concepts: creates complete, self-contained, window-based data management environment in one DOS file. Simplifies everything. Combine functions to create your own solutions in any application: calculations, database management, modeling, text processing, charts, data analysis, statistics, reports, labels, forms, presentations, mail-merge, etc. Simple enough for a PCjr, sophisticated enough for a PC AT. Superb, 400 page hard-cover manual, with many examples. From simple calculations, files, inquiries, to complex models, data structures, reports: when your favorite data manager/ spreadsheet/word processor/integrated system cannot provide the solution you need, remember The Andsor Collection. 60 day money-back guarantee. US$95 + $5 s&h. Visa/MC/ AmEx/Chk/MO/COD. Call or write now to order, or for info. IBM/PC/XT/AT/PCjr, 128K+, one drive or hard disk, monochrome and/or color monitor, DOS 2.0+. Not copy-protected.

The Andsor Collection™

Andsor Research Inc.
181 University Avenue, Suite 1202
Toronto, Ontario, Canada M5H 3M7
(416) 364-8423

SOFTWARE/ COMMUNICATIONS

PC SERIAL DATA ANALYZER

Use your IBM PC or compatible to analyze data streams between two serial devices (up to 9600 BAUD). Two windows display each devices transmission in ASCII or HEX. PC can also act as a terminal for either device. Invaluable tool for debugging serial interfaces. Disk & manual $150.

Triple C Software
2897 SW 13th St.
Fort Lauderdale, FL 33312
(305) 583-0687

SOFTWARE/ DEVELOPMENT TOOLS

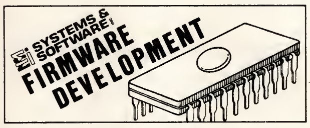
ROMable CODE on PC!

PCLOCATE allows PC users to develop ROM-based software from MS-DOS "Exe" files. The user specifies the physical location of all segments. Output files are compatible with most PROM programmers. PCLOCATE supports the 8086, 8088, 80186, 80188, and 80286 processors. MC/VISA.

ALDIA SYSTEMS INCORPORATED
P.O. Box 37634
Phoenix, AZ 85069
(602) 866-1786

FIRMWARE PRODUCTION ON PC

LINK&LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by C, PL/M compilers & MASM. Provides full control of segments placement anywhere in memory. Supports output of INTEL hex file for PROM programmer, absolute object file for symbolic debugger & ICE, and MS-DOS EXE fifle. Includes an INTEL compatible linker, locator, librarian and hex formatters. $350.



Systems & Software, Inc.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650

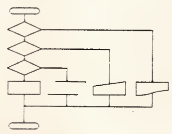
GENSCREEN FOR MS-COBOL

Cobol Source Code Generator for generating the screen section and data division cobol source code for Microsoft and IBM PC cobol. Screen Image Text files are run through GENSCREEN to produce all of the source code for your screen in less than a minute. Super fast programmer productivity tool $69.99.

Personal Computer Development Corporation
P.O. Box 8556
Warwick, R.I. 02888-8556
(401) 333-8704

FLOWCHARTS BY PROFLO

The fast, friendly, flowchart drawing program for the software professional. PROFLO speeds software development and reduces documentation costs. Lets you create, edit, and print clear, precise flowcharts, quickly and easily. Requires IBM PC (or clone) with DOS, 128K, graphics monitor & graphics printer. Program & Documentation $49.95. Demo Disk $10.00.



Star Systems
28 Topstone Drive
Bethel, CT 06801
(203) 744-0546

TURBO PASCAL GENERATOR

GTP APPLICATION DEVELOPMENT SYSTEM, ver. 2. Builds complete, working applications. You give it spec's, it writes error-free code.

Indexed Data Bases	Multiple Screens
Context Sensitive Help	Memory-mapped Video
Global/Search	Full Keyboard Supt

Easy to Use Price $150.00 Visa/MC, ck, MO

AEF
P.O. Box 928
Katy, TX 77492
(713) 391-8570

SCREENWARE UTILITIES

FAST ELEGANT SCREEN CREATIONS FOR PC ASSEMBLY ROUTINES. Make new or edit existing screens, even those CAPTURED in OTHER pgms BY OUR PGM! Supports ALL color/char/ att & monitors. Pop-up menus, help & file management. Creates binary & assembler-ready ASCII source files. Easy to link to main pgm & invokes w/little code. KBD template, charts, 50pg Man, PC/AT-128K, ONLY $27!

Crosswinds Software
8621 Windjammer Drive, Suite M-56
Raleigh, NC 27609
(919) 847-1812

PRODUCTIVITY TOOLS

SRMS™ Software Revision Management System stores all versions of source code in a single library. Allows retrieval of any version of source and application of changes while recording when, why, and where changes were made with no duplication of common code. DOS pathname, directory, and environment variable support, typeset manual, much more.

New version (2.0)................................ $125.00.

QMAKE™ is an intelligent system builder patterned after the UNIX make utility. Only compiles those routines that have changed since last build. Support for macros, multiple entry points, command line parameters. Integrates fully with SRMS™.. $99.00.

MS/PC-DOS 2.0 (MN plus 6%) MC/VISA

QUILT™ COMPUTING
7048 Stratford Rd.
Woodbury, Minnesota 55125
(612) 739-4650

Better BASIC Programmers

BetterTOOLS 2.00 SPEEDS BetterBASIC 1.1/2.0 development. Includes: 190+ useful tools in 17 modules, manual, and source code. No royalties. Quicksort, screen builder, extended math, disk directories, display and printer routines, powerful input, data encryption, on-line error descriptions, much more. Only $89. VISA, MC, COD.

SOFTWARE ASSOCIATES
6220 W. Airport Blvd.
Houston, TX 77035
(713) 726-0706

PC CROSS-ASSEMBLERS

Up to 10,000 lines per minute! Fast X-ref and Linker plus Macros and Librarian. Generates HEX, TEKHEK, S-records, and .OBJ output records. Over 40 micros and XENIX, MS DOS, CPM 80 and ISIS versions. Accepts MOTOROLA and INTEL directives and Mnemonics.

RELMS™
P.O. Box 6719
San Jose, CA 95150
(408) 265-5411

SUPERTECH "PREP" $19.95

Powerful, exceptionally flexible, language-independent pre-processor. Use it, for example, to write register, opcode and operator synonyms, nested macros, line-length identifiers, etc. with assemblers without such features built-in. OK to share. Phone 24 hrs. (Please add $3 s/h, WA res. add 7.9% tax)

SUPERTECH
11410 NE 124 St. Suite 6143
Kirkland, WA 98034-4399
(206) 488-9253

TECH BOOK

SOFTWARE/ DEVELOPMENT TOOLS

SOURCE CODE CONTROL
TLIB™ stores ALL versions of your source in ONE compact library file, even with hundreds of revisions. Updates (deltas), 5-7 times faster than Unix SCCS. Date & comments for each version, easy retrieval. LAN-shared libraries. Free public domain MAKE (with source) by Landon Dyer. DOS 2.x/3.x $99.95 $3 s/h VISA/MC.
Burton Systems Software
P.O. Box 4156
Cary, NC 27511-4156
(919) 469-3068

Fortran Addenda '86
Libraries for graphics and friendly/interactive programs. ASMUTIL2: Total PC control; printers (3), CRTs (2), disks, FULL keyboard, strings, high-speed gets/puts, line/box, fills, tile-painting, EGA graphics modes. BUTILE2: Input word-processing/editing, non-overflowing formats, window management, ... 100 easy to program, "smart" routines + defaults/toggles. 170 pg. manual & annotated samples. $95 alone; both $165. Specify compiler and version.
IMPULSE Engineering, B.R. Strong, Jr.
P.O. Box 3540
San Francisco, CA 94119-3540
(415) 788-4611

The SCREEN GENERATOR
SAVE TIME! Powerful Screen Designer and Memory Resident Screen Manager eliminates tedious screen management tasks from your program. Handles all screen display & data entry. Easy access to Screen Manager from your program. Not a Code Generator! NO Royalties, Not Copy Protected. Easy conversion from other systems. Use with BASIC, TURBO ($59), FORTRAN, C, Realia COBOL, BetterBASIC, PL/M-86, or Assembler. Call if not listed. Price $125.
THE WEST CHESTER GROUP
P.O. Box 1304
West Chester, PA 19380
(215) 644-4206

TREES ENDS DIAGRAM LAYOUT DRUDGERY $29.95!
Decision trees; composition of data flows and data stores; Functional decomposition; Family trees, Process descriptions and structure diagram; Structured English; Warnier-Orr data structure, process structure, and data flow. TREES frees you from positioning symbols!
- Fits brackets as your text grows/shrinks.
- Line wraps and scrolls each text element.
- Diagram stores as machine-readable tree.
- 1000 elements + 32760 characters/diagram.
- Requires MS-DOS, hard disk, 512K.

Ver. 2.0 $29.95 VISA/MC orders: 618-242-4161

Trees™
Box 687
Mt. Vernon, IL
62864

StruBAS- the BEST BASIC!
Adds structure, full-screen handling, indexed files and menus to BASIC. Translates structured BASIC to Microsoft BASIC. Provides interactive screen definition and creates runtime code and tables. Includes subroutine library and sample application. Not copy protected. $150 + $5 s/h. Demo $5. VISA/MC
LANEY SYSTEMS, INC.
3 Office Park Dr. Suite 100
Little Rock, AR 72211
501-225-7755

Documentation Generator
Customizing Documentation Generator offers:
FLEXIBILITY—define your documentation
SIMPLICITY—follow DOS tree structure
CREATIVITY—run all commands and programs
COMMAND ABILITY—repeat an expanded command
I/O ABILITY—run special I/O redirection
check $39, MO, VISA/MC.
U-NEED & SEE COMPUTER SYSTEMS
P.O. Box 7363
Thousand Oaks, CA 91359
(805) 495-8441/(818) 889-3324

METRICS FOR MANAGEMENT
ANALYZE reads your code and produces metrics on VOLUME, COMPLEXITY, PRODUCTIVITY & more! Management, QA/QC & Engineers gain valuable insight into scheduling, problem areas, testing, maintenance & compliance to standards. For IBM PC. Process FORTRAN & "C" $495.00 + S/H. Add Ada for only $200.00 more! MC/VISA
AUTOMETRIC INC.
891 Elkridge Landing RD., Suite 350
Linthicum, MD 21090 (301) 859-4111

Tools for CB80 & CB86
BDOS, DOS, and BIOS calls from CB80 and CB86! CBC Tools includes functions for directory access, string ops, a debugger, radix conversion, command line parsing, quicksorts, bit and byte ops, and more for CP/M-80, CP/M-86, and PC-DOS, $180.00. Ask about our Pascal MT+ and Quickbasic products.
Minnow Bear Computers
P.O. Box 2233 Sta. A
Champaign, IL 61820-8233
(217) 398-6883

IBM® PC MANAGEMENT TOOLS™
• Forecasting • Inventory Control • Quality Control • Project Mgt. • Statistics. • Plant Layout • Financial Mgt. • Production Planning. 40+ New programs (not pub. dom.) w/544pg. user manual. FREE BASIC SOURCE CODE Not Copy Protected! Visa, MC, Amex, Cks & Ppd PO's. $99.95 + $7.50 s&h +5% GA tax. Volume Discounts! Call or Write.
MANAGEMENT INFORMATION SYSTEMS, INC.
Dept. AA P.O. Box 98209
Atlanta, GA 30359, (404) 231-1297

C PROGRAMMERS, IT'S NEW
Total screen development facility. Compose toughest screens in minutes!! Let EZ-Screens save you time and $. Full screen I/O control. Switch selected complete debug feature. Powerful formatted I/O routines. Easy to use. Natural commands. Manual included. $49.95 VISA/MC/CHECK. 30 DAY $back guarantee.
Commonwealth Computer Corporation
Box 407
Weymouth, MA 02188
(617) 848-3180

SOFTWARE/ EDUCATIONAL

THE DGI WORDSEARCH PUZZLE
A great teaching aid to build vocabulary, improve spelling, or for your own enjoyment. Up to forty words can be used. Size can be 20 x 20 or 20 x 40 characters. Each puzzle has a unique solution. If you print the same puzzle twice each solution is different. The DGI NUMBERSEARCH is also included. For the IBM PC or compatible computers. $30.
DECISION GRAPHICS INC., P.O. Box 2776-PCTJ
Littleton, CO 80161, (303) 796-0341

SOFTWARE/ ENGINEERING

METAL FABRICATORS
PC/Cutlist takes input from your bill of material—Detail drawing and calculates the best cutting combination for any length stock and prints a shop ready cutting list and scrap report. Also an optimization feature finds best mult length for mill orders. Price $300. Demo Disk $25.00
THE JOSEPH ALBERT CO.
P.O. Box 611
Blue Island, Illinois 60406
(312) 349-9032

PC/8087 VECTOR PROCESSING
The VECTOR87 library is written in assembler, includes 60 routines to turbocharge your math & scientific programs. FFT, vector operations, convolution, etc. 1K real FFT takes only 1.2 second. Versions for Fortran (MS, R/M, Lahey), C (MS, Lattice). $120 each version with source, no royalties. Write for technical information.
VECTORPLEX Data Systems Ltd.
P.O. Box 138, Station M
Calgary, Alberta, Canada T2P 2H6
(403) 248-1250

ENGINEERING DESIGN SOFTWARE
20+ hardworking programs for Mechanical Engineers, Electrical, Energy, Hydraulic, Fire Sprinkler, Structural.
NEWLY UPDATED:
- Fault Current
- Manual "J" Clg/Htg 7th Ed
- Tree Fire Sprinkler Design

Primary developers of engineering software.
MC2 Engineering Software
STE 425 • Box 430980
Miami, FL 33143 • (305) 665-0100

Filter Design
Digital Filter Tutor, $375. Kalman Filter Tutor, $850. Practical hands-on training courses. Learn how to specify, design, tune, implement, analyze, and test filters. Ideal alternative to text books, seminars, and university courses. FREE demo disk available.
Engineering Tutorial Software
22338 Lull Street
Canoga Park, CA 91304
(818) 716-0816

SOFTWARE/ FILE MANAGEMENT

Wilkes Database
For IBM-PC and compatibles. Paint screens to create/view/update/delete/sort/search/print/reformat/arithmetically manipulate new or extant files. ASCII, 16/32 bit integer and 32/64 bit floating point storage. Powerful key sort and report formatting. $325. Demo $40.
Wilkes Software
5231 Longwood Drive
Memphis, TN 38134
(901) 386-4712

SOFTWARE/GENERAL

PUBLIC DOMAIN SOFTWARE IN C
Over 60 volumes of public domain software in CP/M & MS-DOS formats.
- editors & compilers
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details.

C Users' Group

THE C USERS' GROUP
P.O. Box 97
McPherson, KS 67460
(316) 241-1065

SOFTWARE/GRAPHICS

TEKTRONIX 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

SCIENTIFIC DATA PLOTTING
SCI-GRAF creates graphs up to 1680 x 1712 dots (over 3 million pixels!) on Epson or IBM graphics printers. Supports log scaling, overlays, point-labeling, legend creation, batch mode, wide-carriage printers, and color graphs on a JX-80. Requires DOS 2 or 3, 256K. No credit cards. $99.95
Microcomputer Systems Consultants (MSC)
32 W Anapamu Suite AD
Santa Barbara, CA 93101
(805) 966-9247

TECH BOOK

SOFTWARE/GRAPHICS

COMPUTER AIDED GRAPHICS
Free form 2D full screen graphics editor for the IBM-PC and compatibles. Use it to create charts such as: Organization, Gantt, Flow and Pie. Also produces Graphs, Work area plans, Electronic schematics and more. Drives HP or compatible plotters. Only $49.95 a price of a game. For more details send a long SASE to:
WorkWare
P.O. Box 661295
Sacramento, CA 95866-1295
(916) 922-5819

MetaWINDOW™/TurboWINDOW™
Advanced graphics toolkit provides Xerox Star/Apple Macintosh style graphics on your IBM PC. Supports most popular graphics cards. Allows you to create pop-up menus, windows & icons; use proportionally spaced fonts; rubberband & drag lines, text or bitmap images; supports mouse-cursor tracking. Tightly optimized for use with Turbo Pascal, IBM Pascal, C, Fortran.
METAGRAPHICS SOFTWARE CORP.
4575 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-1550

FORTRAN GRAPHICS LIBRARY
GRAFMATIC (screen graphics): 75 MS FORTRAN/Pascal, R-M/Profort, Lahey FORTRAN callable subroutines. Fully documented, prof. graphics capabilities, inc. general utility, 2-D interactive, total 2-D plots, 3-D plots and solid models. $135. H-P or H-I plotter? get PLOTMATIC, complete plotter graphics library. Interfaces w/GRAFMATIC. $135. Both $240.
MICROCOMPATIBLES, INC.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

FORTRAN GRAPHICS LIBRARY
MicroGlyph's SciPlot provides MS V3.2/F77L1 IBM Professional FORTRAN graphics library on IBM PC/XT/AT under PC DOS. Features high resolution plots from printers (IBM/EPSON/Apple LaserWriter), or graphics displays (IBM Color/HERCULES-Monochrome/TECMAR-GM). Price: $69.95 + $3 shipping + 4% MASS tax.
MicroGlyph Systems
P.O. Box 186
Lexington, MA 02173

SOFTWARE/LANGUAGES

EASY WAY TO MASTER DOS!
New handbook shows how, why, and when to use all 44 PC & MS DOS commands quickly and easily! Lively hands-on tutorial makes learning fun, for a change. Covers basic thru advanced topics. All versions up to 3.1. Lots of practical examples. 221 pages only $15.95 at bookstores, or send check + $1.00 S&H to:
LITTLE, BROWN & COMPANY PUBLISHERS
200 West Street, Code T-368121
Waltham, MA 02254
1-800-343-9204 (credit cards)

CORRECTFORTH
Version 1.1 has a full screen editor & generator. 8086-186-286 assembler with support for the 8087/286/287 coprocessors. Cross assemblers for 8080 & 6502 included. 83 standard DTC. Debugging utilities for multiple processes. Fully MS-DOS compatible. $80.75. Visa/MC/Chk/MO/Amex.
Correct Software Inc.
RRI, Box 140
Black Hawk, SD 57718
(605) 787-5904

FINALLY!
A Subroutine Library for Compiled BASIC load directories into arrays, WINDOWING, graphics pie and bar charts, special string functions; 130 more routines to add power to Microsoft QuickBASIC or IBM BASIC Compiler 2.0. 30 Day Money-Back Guarantee.
VISA/MC/Check/MO
FINALLY! is only $99.00 + $4.00 S/H
KOMPUTERWERK, INC., Dept PCT
851 Parkview Blvd.
Pittsburgh, PA 15215
(412) 782-0384

Simulation Language
TurboSim allows large discrete event simulations i.e., manufacturing systems, health-care systems, etc. to be run on a micro. It features a comprehensive final report, histograms, plots and multiple simulation runs in an unattended mode. Source code is written in Turbo Pascal and is included. Borland International's Turbo Pascal and an IBM/PC with 64K are required. $49.95.
Micro Simulation
37 William J. Heights
Framingham, MA 01701
(617) 875-6098

SOFTWARE/NETWORKING

LAN SOFTWARE
LAN software for your PC, XT or AT, allows you to connect to your VAX and other systems. Provides file transfer and terminal services at hundreds of kilobits/sec. TCP/IP or XNS protocols available; MS-DOS, XENIX, VENIX and PC/IX supported, many additional options, basic PC package only $350.

Network Research Corporation

NETWORK RESEARCH CORP.
2380 North Rose Avenue
Oxnard, CA 93030
(805) 485-2700, TLX 297579

SOFTWARE/PUBLIC DOMAIN

TURBO PASCAL™ SOFTWARE $6
Write or call for information about:
- Systems & applications development tools
- Programs for home and business
- Communication tools & applications
- Games in specialized applications
- Scientific/engineering programs & routines
- Graphics including animation tools

TURBO S.I.X.
P.O. Box 8373
Waco, TX 76714
(817) 753-2182

THE BEST OF THE BEST!
Public Domain & User Supported Software for IBM PC & Compatibles! Wordprocessing, Accounting, Spreadsheets, Database, Modem, Games, Languages, etc., etc. 50 disks crammed Full-$205.00! or rent for 2wks. $75.00. Info. and Super Sampler Disk $6.50. Deluxe Word Processor $6.50. Both $12.00. MC/VISA.
BLUE CIRCLE GROUP, Inc.
P.O. Box 23502; Dept. TJ
Minneapolis, MN 55423
(612) 823-4111

SOFTWARE/SCIENTIFIC

SCI/ENG GRAPHICS
OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

SOFTWARE/SECURITY

"NEW" BIT-LOCK® SECURITY
Piracy SURVIVAL 4-YEARS proves effectiveness of powerful multilayered security. Uses rapid decryption algorithms and small reliable port for transparent security device. NOW AVAILABLE for PARALLEL or SERIAL port. NEW KEY-LOK™ security device available at HALF-PRICE.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 798-7683 or 922-6410

SECURE AT/XT/PC
Control system access, data access! FiXT/S. Control system boot for most popular XT/PC hard disk controllers. Feature for AT-and-XT-compatible HD controllers segments hard disk by volumes, controls access with passwords, supports hard disk expansion. $80 $120 + $3 shpg. plus CA tax.

Golden Bow Systems
2870 Fifth Ave. Suite 201
San Diego, CA 92103
(619) 298-9349

SMART COPY PROTECTION
In the past, Copy-Protecting your programs meant inconvenience to the end-user, limited or no hard disk support, and a lot of added expense per product copy. Our new software based EVERLOCK System solves these problems. For IBM and compatibles, call for info.

AZ

Az-Tech Software Inc.
423 Grandview
Richmond, MO 64085
(816) 776-8153

SOFTWARE/SERVICES

TAPE/DISK CONVERSIONS
Conversion services to or from over 500 computer systems:
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
Pivar Computing Services, Inc.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

RENT POPULAR SOFTWARE
Finally, you can rent popular software for your IBM or compatible. See how good that game really is. Try that business package before dishing out huge amounts of money. Not public domain. No membership fee. Where else can you get $75.00 programs for $5.00? Send $1.00 for complete catalog.
R.S.D.
Dept. TJ
P.O. Box 272
Bronxville, NY 10708

TECH BOOK

Software/Statistics

RATS!
RATS is the ultimate econometric software package. It is powerful, fast, accurate, and inexpensive. RATS performs regression analysis, time-series forecasting (including Box-Jenkins) and cross-section (including logit and probit). RATS supports a variety of data formats, including Lotus WKS files. RATS also offers a programming language that provides maximum flexibility and power. Spectral analysis and graphics are available as options. Base program $200. Demo including 300 page manual $40. MC/VISA.
VAR Econometrics
P.O. Box 19334
Minneapolis, MN 55419
(612) 822-9690

STATISTICS FORECASTING
TWG/ARIMA—a univariate Box-Jenkins forecasting package, designed for statisticians.
EASI/ARIMA—same as above, for the non-statistician. $450. each.
ELF—The Statistical Package—a general purpose statistical package. $350. Call or write for more information.
THE WINCHENDON GROUP, INC.
P.O. Box 10339
Alexandria, VA 22310
(703) 960-2587

STATISTIX™—ONLY $75!
STATISTIX is a powerful and very easy-to-use interactive statistical system for micros. Used by many major universities, businesses, state governments and research organizations. Please check us out before you buy a statistics program; you'll agree SX is a "best buy"! SATISFACTION GUARANTEED—For more info:
NH ANALYTICAL SOFTWARE
801 West Iowa Avenue
St. Paul, MN 55117
(612) 488-4436

Software/Terminal Emulation

Color VT 102 & 4010
PC102 turns an IBM PC/XT/AT/jr to a VT 102/52 terminal with printer/ASCII & KERMIT file transfer support. Optional: 132 col. 4010 graphics support. Guaranteed compatibility with all VT 100 applications including EDT, WORD11, All-In-One, & UNIX. Easy to use. From $89. Call for free package!
General Micro Systems
P.O. Box 5330, Dept. TJ
Hopkins, MN 55343
(612) 944-0593

SIDETERM
Pop-up VT200/100/52 emulation package accessible from inside any program. Window can be moved and sized at any time. Data can be cut/pasted between terminal and application. Virtual 132 col. screen. Supports EDT, UNIX, and scrolling regions. Not copy protected. $59.99. CK/MO. Demo $10. MO res. add 6.225% tax.
SOFTWARE INNOVATIONS
P.O. Box 36422
Kansas City, MO 64111
(816) 756-2745

Software/Utilities

AT/XT/PC HARD DISK EXPANSION
"Replace hard disk with a bigger one, or add a second drive! Vfeature BREAKS THE 33 MBYTE BARRIER on standard AT, XT, and compatible hard disk controllers. Includes multiple volumes, security features, selectable clusters, keyboard lock. $80-$120 + $3 shipping + CA Tax"

Golden Bow Systems
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed, req. IBM PC, XT, AT, DOS 1, 2, 3, 192K + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept TB2
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

CHARACTER CUSTOMIZATION
CHARGENI 2.1 works with the IBM EGA to let you modify the character set, allowing many word processors to display technical material, equations or other special characters. Also works on the Color/Graphics Adapter in graphics modes. Requres DOS 2.x or 3.x, IBM Standard or Enhanced Graphics Adapter. $35 + $2 s/h (MN add 6%).
DK Micro Consultants
P.O. Box 6714
Minneapolis, MN 55406
(612) 722-0931

TallScreen—DOS POWER
Natural extension of DOS. Scroll back through screen output, edit text on full screen, mark blocks to printer or file, recall commands & directories, enter multiple commands, capture screens from application programs, create user profiles. Solid tech support. A real bargain at $49.95 VISA/MC

QUALITAS™

Qualitas, Inc.
8314 Thoreau Drive
Bethesda, MD 20817
(301) 469-8848

HELL uva SHELL
Most DOS interfaces are designed for IDIOTS! Get the BEST full-screen interactive directory around ... DOS command recall ... command aliasing ... multiple commands at DOS prompt ... find files anywhere on disk ... find complex text patterns in files ... browse files ... copy files based on time stamp & MORE. Only $39! DOS 2.0+, uses 96K memory. NCP!/VISA/MC
Dwarf Nebula Software
666-SHELL
P.O. Box 46, Dept. TJ
Sugarland, TX 77487
(713) 980-7355

TurboRef 4.0
Pascal programmers get organized with TurboRef! TurboRef will cross reference a source program and create a program listing. TurboRef can process a list of files, will read "include" files, identifies line number for each reference. New release now avail. IBM PC and jr., XT, AT or compatible, only $49.95; VISA/MC, or check.

GRACON SERVICES, INC.

GRACON Services Inc.
4632 Okemos Rd.
Okemos, MI 48864
(517) 349-4900

LIST-IT
Make your IBM BASICA and IBM compiled Basic program readable with a pretty print. Cross references for statements and variables are included. While-wend, FOR-NEXT and IF-THEN-ELSE structures recognized. $29.95. Calif. residents add 6%. CHECK/MO/VISA/MC
PCS
50 Cornell #9
Irvine, CA 92715
(714) 854-1282

SAVE THAT SCREEN!
Do you immediately reach for the PrtSc key to save screen info? What a waste of time and paper! Now, SCREENSNAP™ lets you save and recall up to 9 screens at the touch of a key. Friendly with other resident programs but unlike some it is compact; will run in as little as 5K. Also includes useful utilities to save and recall from files, programmer's interface and sample code. Build your own help screens with your text editor, then save and recall them with SCREENSNAP. $39.
Aldridge Real-Time Software Inc.
Call 800-443-4160; NJ (201) 846-7242

FILE PRINT MANAGER
GLISTER™
★ Use DOS wildcards to build a list of up to 100 files to print
★ Save/restore file lists
★ Restart a file on any page after a printer jam
★ Print multiple copies
★ Control: margins, line/page length, spacing, user-formatted header/footer lines and more
★ Prints files as fast as printer is capable $49
Aldridge Real-Time Software Inc.
Call 800-443-4160; NJ (201) 846-7242

REPEAT PERFORMANCE™
OUTSMARTS IBM PC KEYBOARD LIMITATIONS! Smoothly ACCELERATES CURSOR MOVEMENT, speeds up editing. High speed scrolling. You set ADJUSTABLE REPEAT SPEED and DELAY. SKID SQUELCH™ feature auto-adjusts to match performance of word-processors/text editors/spreadsheets. Cursor & scrolling STOP INSTANTLY when key is released. EXPANDS TYPE-AHEAD BUFFER-type DOS comamnds while your programs compile. Less than 4K resident. WARNING: ADDICTIVE! IBM/PC/XT/AT compatibles. $back guar. $39.95 +$5 p&h. VISA/MC/AMEX/CK. Site licensing from second copy! Limited offer outstanding.

popular demand™

Popular Demand Inc.
62 South 1025 East
Lindon, UT 84062
(801) 785-0101

AUTOMENU™ VERSION 4.0
Create one menu system to run all your programs, batch files and DOS commands. "Insulates" novices; many options for power users. Onscreen help, password protection, user-defined prompts. Written in assembler. 16K size. Over 7,000 satisfied users. Money back guarantee. $46 + $4 s/h. Chk/Visa/MC.

MAGEE ENTERPRISES

Magee Enterprises
6577 Peachtree Industrial Blvd., Dept. T9
Norcross, GA 30092-3796/USA
(404) 446-6611

VCACHE GETS YOUR DISK MOVING!
Hard disk accelerator increases speed of cartridge and fixed disk operations using memory caching to eliminate repetitive disk access. Allocate up to 15Mb of extended or expanded memory, or .5Mb of standard memory for caching disk data. Includes diskette and screen accelerator modules. Automatic and transparent after installation. $65+ $3 shpg, CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

TECH BOOK

SOFTWARE/UTILITIES

HARD DISK EXPANSION!
Disk Manager allows the installation of any ST506 hard disk on PC,XT,AT and compatibles. Volumes up to 256mb! Menu driven/auto install, compatible w/ all vers of MS/PC DOS (does not modify DOS), up to 16 volumes, easy to use! $125+ ship. Ask about Novell product! Dealer inquiries invited.

ONTRACK COMPUTER SYSTEMS INC.

Ontrack Computer Systems, Inc.
7460 Washington Ave. S.
Eden Prairie, MN 55344
(612) 941-4504

DISK UPGRADE BIOS for ATs
DUB-14 overides AT Drives Table to allow any compatible drive to be attached and fully used on the standard AT controller. Two ROMs plug into empty sockets on system board. Includes complete Set-Up routine and low-level format facility. Works with UNIX, XENIX, other OS and networks. $95 + $3 shpg. CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

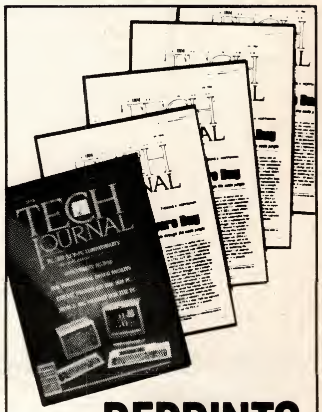
RUN TURBO FROM BATCH FILE
Now you can do any of the things in the Turbo Pascal environment, except edit, from a batch file. Requires Turbo Pascal and DOS 2.0 or higher. Price $14.95. For an additional $10.00, will send the source code, plus a surprise! Add $2.00 for S/H.
DONALD E MYERS
10051 SW 70th Street
Miami, FL 33173
(305) 596-9284

SOFTWARE/WORD PROCESSING

FORMATH™ TEXT-FORMATTER
For technical texts. Sizes & places equations, matrices, ratios, integrals, big symbols, footnotes. Macros, multiple fonts, 158 Greek/math symbols. Automatic hyphenation, section/equation/reference numbering, indexes, table of contents. Dot-matrix, daisywheel, laser printers, plain/graphics monitors. $400. $50 for demo.

$$\lim_{n\to\infty}\sum_{i=1}^{n}\begin{bmatrix}\frac{1}{i(i+1)} & \frac{1}{2^i}\\ \frac{(-1)^i}{2^i} & \frac{1}{\pi^2 i^2}\end{bmatrix}=\frac{1}{2}\begin{bmatrix}\int_0^1 \frac{\log(1-x)}{-\pi^2 x}\,dx & -1\\ \frac{1}{3} & 1\end{bmatrix}^{-1}$$

SHANTHA SOFTWARE, INC.
50 West 97th St. Room 11N
New York City 10025
212-222-7647
Touch-tone Toll-free: 950-1088-wait-FORMATH

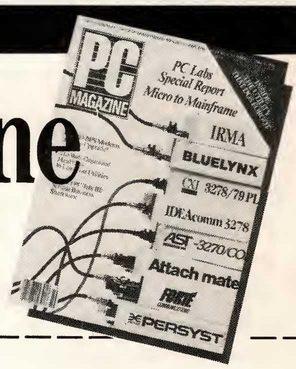


REPRINTS AVAILABLE

Quantity reprints of articles appearing in **PC Tech Journal** are available and will be prepared to meet any special requirements. Inquiries should be directed to Eileen Pfeiffer, Reprints Dept., Ziff-Davis Publishing Co., 1 Park Ave., New York, New York 10016. Phone 212-503-5447.

PC TECH JOURNAL

Subscribe to PC Magazine and save 65%



You'll receive in-depth and comprehensive product evaluations and comparative hardware and software evaluations from PC Labs, the publishing industry's *only* testing laboratory.

You'll receive new applications, power tips and special reports designed to add more power and flexibility to your IBM/MS DOS-standard system.

And you'll network with other users and discover routines and techniques that can save you time and money.

Subscribe to *PC Magazine* today! Send the coupon to: *PC Magazine* · P.O. Box 2443 · Boulder, Colorado 80322 *or call Toll-Free; 1-800-852-5200 now!*

Yes, send me PC Magazine for:
☐ 22 issues (one year) for $24.97. Save 62%!
☐ 44 issues for only $44.97. Save 65%!
Savings based on annual single-copy price of $64.90.

Mr./Mrs./Ms.________________________
please print full name

Company________________________

Address________________________

City______________State________Zip__________

Add $15 per year for postage outside USA, US currency only.
Please allow up to 60 days for delivery of first issue. Basic annual subscription price is $34.97.

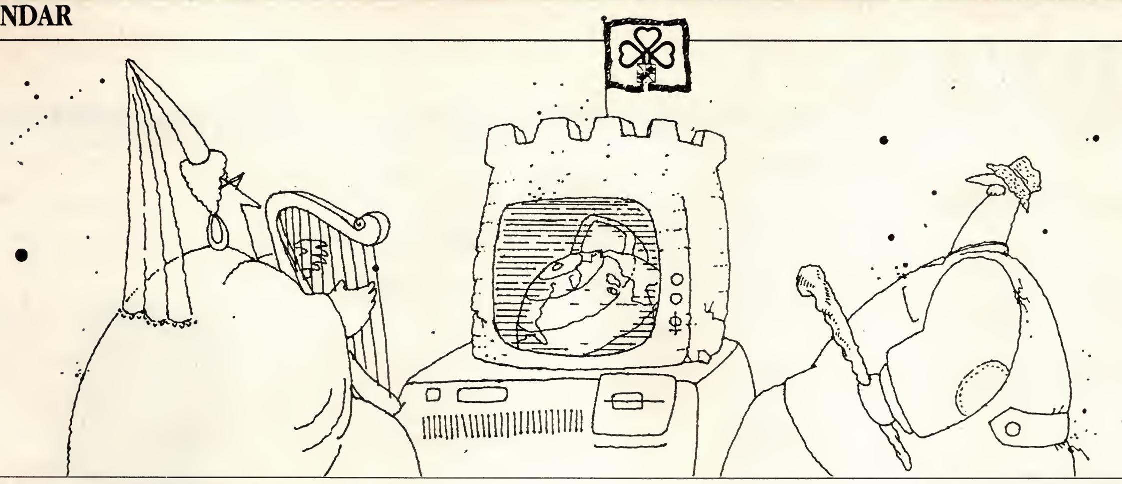


ILLUSTRATION • DAVID POVILAITIS

SEPTEMBER

September 1-5
World Computer Conference
Dublin, Ireland
Sponsor: International Federation for Information Processing
Contact: IFIP Congress '86, 1899 Preston White Drive, Reston, VA 22091; 703/620-8900

September 8-10
NCC Telecommunications Conference
Philadelphia, PA
Sponsor: AFIPS, ACM, DPMA, IEEE, SCS
Contact: Mike Sherman, AFIPS, 1899 Preston White Drive, Reston, VA 22091; 703/620-8935

September 8-10
1986 ACM Conference on Research and Development in Information Retrieval Systems
Pisa, Italy
Sponsor: ACM SIGIR and AICA-GLIR
Contact: L. Rossi Bernardi, C.N.R. P/zza A. Moro 7, Roma, Italy

September 15-18
Euromicro '86: Symposium on Microprocessing and Microprogramming
Venice, Italy
Contact: Mrs. Chiquita Snippe-Marlisa, p/a TH Twente, Department INF, Room A227, P.O. Box 217, 7500 AE Enschede, The Netherlands; +(31) (53) 338799

September 16-18
NetWorld
Dallas, TX
Sponsor: Novell, Inc.
Contact: 1170 N. Industrial Drive, Orem, UT 84057; 800/LANFAIR

September 30-October 2
Northcon '86
Seattle, WA
Sponsor: IEEE
Contact: Northcon '86 Professional Program Committee, Dale Litherland, Director of Education, 8110 Airport Blvd., Los Angeles, CA 90045; 800/421-6816, in California, 800/262-4208

OCTOBER

October 5-10
International Conference on Computer Design
Atlantic City, NJ
Sponsor: ICCD
Contact: ICCD '86, 1730 Massachusetts Avenue NW, Washington, DC 20036-1903; 202/371-0101

October 15-17
Scan-Tech '86
San Francisco, CA
Sponsor: Automatic Identification Manufacturers, Inc.
Contact: Automatic Identification Manufacturers, Inc., 1326 Freeport Road, Pittsburgh, PA 15238; 412/963-8588

October 16
Communications Network Procurement
Atlanta, GA
Sponsor: Georgia Institute of Technology
Contact: Deidre Mercer, Department of Continuing Education, Georgia Institute of Technology, Atlanta, GA 30332-0385; 404/894-2547

October 20-22
UNIX Expo
New York, NY
Contact: Don Berey, National Expositions Company, Inc., 49 W. 38th Street, Suite 12A, New York, NY 10018; 212/391-9111

October 20-24
Using Computer Graphics to Enhance Productivity and Management Effectiveness
Atlanta, GA
Sponsor: Computer Graphics Atlanta
Contact: CGA '85 Conference Director, 2033 M Street NW, Suite 333, Washington, DC 20036; 202/775-9556

October 25-26
Information Systems Education Conference (ISECON) '86
Atlanta, GA
Sponsor: Data Processing Management Association Education Foundation
Contact: Dr. Kathy Brittain White, University of North Carolina, Information Systems, Greensboro, NC 27412; 919/379-5666

October 26-November 3
INDIA '86
India
Sponsor: Industrial and Trade Fairs International Limited
Contact: Mr. Hasu Ramji, Industrial and Trade Fairs Limited, Oriel House, 26 The Quadrant, Richmond upon Thames, TW9 1DL, UK; 01-940 6065

NOVEMBER

November 2-5
Fall Joint Computer Conference
Dallas, TX
Contact: Roberta Bukar, ACM Headquarters, 11 W. 42nd Street, New York, NY 10036; 212/869-7440

CALLS FOR PAPERS

Deadline: September 10
Conference on Human Factors in Computing Systems and Graphics Interface
Toronto, Canada
(April 5-9, 1987)
Sponsor: ACM/SIGCHI
Submit papers to: Peter Tanner, Computer Systems Research Institute, 10 Kings College Road, University of Tornoto, Toronto, Ontario M5S 1A4; 519/888-4589

Deadline: October 31
International Workshop on FORTH and its Applications
Taipei, Republic of China
(October 31-November 2)
Sponsor: FORTH Interest Group
Submit papers to: FORTH Interest Group, P.O. Box 8231, San Jose, CA 95155; 408/277-0668

"ADVANCED LOGIC SHIPS AT™ COMPATIBLE WITH AN XT™ CLONE PRICE"

Computer Retail News
Monday, July 7, 1986

The newest generation of personal computers are designed and manufactured by ADVANCED LOGIC RESEARCH, INC. Heir to all the best traits of its ancestors, the PC2/286 series of personal computers possess total compatibility to past and current hardware and software achievements.

SMARTER DESIGN!

Like all products in the ALR line, the PC2/286 is born with enhancements and the latest technology available. THE NEXT GENERATION has speed and power equal to that of the AT™, is as small as the XT™, but the complete system (including graphics video card and high resolution monitor) costs less than both! How? Smarter design! Like we've always said ALR doesn't design products without improving them beyond what our competition offers.

Smarter design, as well as being superior, is less expensive to manufacture so you pay less.

We've incorporated all the knowledge that the past successes and failures of big manufacturers have afforded and used it to make a better product.

THE NEW STANDARD

The evolutionary PC2/286 brings the power of 80286 (AT) technology into a price range that until now has been the exclusive domain of the PCs. The PC2/286 is destined to be the new standard in a rapidly changing marketplace and is designed to meet the ever increasing needs of today's users.

PERFORMANCE

PC2/286
$1545.00

- 8 MHz AT compatible 80286-8 CPU
- 2 AT slots, 3 PC slots
- 512 KB RAM standard expandable to 1 MB on board
- 1.2 MB floppy disk drive
- 1 serial, 1 parallel printer port
- Hercules compatible Monographics Adapter, AT compatible floppy disk controller on board
- 130 Watt UL/CSA Power Supply
- Clock/Calendar Battery Backup
- AT style 83 Keyboard
- High resolution Monochrome Monitor

PC2e/10MHz

- 10 MHz 80286-10 CPU 8-10 MHz switchable
- 20 MB/30 MB 40ms high speed hard disk
- 1.2 MB high density floppy disk drive
- 16 Bit high speed hard disk controller
- Memory expanded to 1 MB on board
- 1 serial, 1 parallel printer port
- Hercules compatible Monographics Adapter
- 12'' high resolution Monochrome Monitor
- EGA Video card and Multi-Sync Monitors available

Advanced Logic Research, Inc.
10 Chrysler, Irvine, California 92718 - (714) 581-6770
FAX: (714) 581-9240 - TELEX: 5106014525, Answer back Advanced Logic
IN CANADA CONTACT ALR (416) 229-6477

CIRCLE NO. 116 ON READER SERVICE CARD

See Us at COMDEX, Las Vegas Booth M-1218, MGM Grand Ballys

Remote lets you run almost any program, from any location, as if you were there.

Remote is the software that turns your personal computer into a host computer. You or anyone you choose can dial it up from almost any terminal in almost any location, and run most popular application programs such as word processing, spreadsheets, and data base managers.

You'll see the program on your remote terminal screen as if you were seated at the host PC.

While Remote itself becomes transparent in use, it offers some very tangible benefits:

- You don't need a second PC to do the job of two. Almost any terminal or terminal emulator will do. The only software you need is the software in your host PC.
- Each of several different users can call in from anywhere in the world and use the host PC and software. Remote includes a sophisticated electronic mail system with encrypted messages and individual password protection.
- You can transfer files to and from the host computer, using the Crosstalk or XMODEM protocol.
- Programmers and software vendors can use Remote to debug a client's software by phone, without leaving their own offices.

Imagine the potential Remote has in extending the power of your own PC. Ask your dealer about it, or write for details.

REMOTE™

MICROSTUF®
1000 Holcomb Woods Parkway
Roswell, Georgia 30076

CIRCLE NO. 167 ON READER SERVICE CARD